Mentes, máquinas y matemáticas

La inteligencia artificial y sus retos

Ignasi Belda

El mundo es matemático

© 2011, Ignasi Belda por el texto
© 2011, RBA Coleccionables, S.A.

Realización: EDITEC
Diseño cubierta: Llorenç Martí
Fotografías: iStockphoto, Corbis, Getty Images

Reservados todos los derechos. Ninguna parte de esta publicación puede ser reproducida, almacenada o transmitida por ningún medio sin permiso del editor.

ISBN: 978-84-473-7436-6
Depósito legal: NA-2692-2011

Impreso y encuadernado en Rodesa, Villatuerta (Navarra)

Impreso en España - *Printed in Spain*

Sumario

Prefacio 7

Preámbulo 9

Capítulo 1. ¿Qué es la inteligencia artificial? 13
El test de Turing 13
La búsqueda 19
El aprendizaje 22
La planificación 24
El razonamiento automático 25
El procesamiento del lenguaje natural 32
Para terminar, gestionar lo que se sabe 33

Capítulo 2. Búsqueda 37
Darwin ya lo dijo 38
La inicialización 41
La evaluación 41
La selección 43
La reproducción 46
El reemplazamiento 49
Un ejemplo práctico: evolucionando hacia un buen fármaco 50

Capítulo 3. Aprendizaje artificial 55
Un ejemplo de aprendizaje: la predicción de tumores 56
Otro ejemplo: el marketing online 61
El cerebro del robot: las redes neuronales 64
Las neuronas se agrupan 67
Y el cerebro funciona 72
Y el cerebro se complica 73
¿Son necesarios los exámenes? 75

Capítulo 4. Planificación y razonamiento automático 81
Cómo se gestiona un trasplante 81

Planificar, ésa es la palabra 85
Detección de conflictos 90

Capítulo 5. Análisis de datos 95
La minería de datos 96
La maldición de la dimensionalidad 99
Visualización de datos 104
Reconocimiento de patrones 106
Un ejemplo práctico: análisis de ventas 107

Capítulo 6. Vida artificial 111
Introducción a la vida artificial 111
Sistemas adaptables complejos 114
Primera propiedad: la agregación 116
Primer mecanismo: el etiquetado 116
Segunda propiedad: la no-linealidad 117
Tercera propiedad: la formación de flujos 119
Cuarta propiedad: la diversidad 121
Segundo mecanismo: los modelados internos 123
Tercer mecanismo: los bloques de construcción 124
Los autómatas celulares 125
Sistemas inmunes artificiales 128
Inteligencia de enjambre 131
Aplicaciones de la vida artificial 133
Teoría de juegos 133
De nuevo el *data mining* 134
Programación de robots 136

Epílogo 139

Anexo. Conversación con Eliza 141

Glosario 143

Bibliografía 147

Índice analítico 149

Prefacio

No tengo miedo a los ordenadores.
A lo que tengo miedo es a la falta de ellos.
Isaac Asimov

Desde hace varias décadas la inteligencia artificial ha despertado la pasión de muchos científicos, estudiantes y ciudadanos. ¿Qué hay de realidad y qué hay de ficción en las películas, series y novelas en las que a menudo vemos robots parlantes, máquinas autónomas y sistemas automáticos capaces de comportarse como un auténtico ser humano?

A lo largo de este libro intentaremos desentrañar algunos de los misterios que aglutinan estas dos palabras: «inteligencia artificial»; no contentos con ello, también haremos una breve introducción al concepto de «vida artificial». Pues... ¿veremos en el futuro próximo máquinas parlantes? ¿Y a medio plazo? ¿A cuánto estamos de poder interactuar con sistemas autónomos e inteligentes que puedan extraer energía de la comida tal y como lo haría un ser vivo? ¿Es todo ello una ficción?

Todas estas cuestiones y muchas más son las que revisaremos a lo largo del libro. Y más concretamente, analizaremos la inteligencia artificial desde sus cuatro ámbitos principales de aplicación: la búsqueda, el aprendizaje, la planificación y el razonamiento automático. Además, en el capítulo 5 hablaremos sobre el análisis de datos, uno de los campos de mayor uso y aplicación de las herramientas inteligentes, de gran importancia en la era digital que estamos viviendo, en la que segundo a segundo se generan miles de millones de datos que serían inútiles si no existieran las herramientas inteligentes necesarias para poder extraer conocimiento de ellos.

Por último, en el capítulo 6 nos daremos una rápida zambullida en los abismos de la vida artificial. ¿Qué es un ser vivo? ¿Y un ser vivo «artificial»? ¿Tendremos algún día la capacidad de crear organismos vivos e inteligentes con los que poder interactuar? ¿Acaso somos capaces de hacerlo ya? Todas estas preguntas hallarán respuesta mediante ejemplos prácticos del día a día, en los que podremos reconocer los seres «vivos artificiales» que nos rodean ya en nuestra vida cotidiana.

Para terminar este prefacio, sólo nos queda señalar que aunque las ciencias de la computación son una disciplina aplicada de las matemáticas, nos hemos esforzado en gran medida en «ocultar» la vertiente más matemática y analítica que da soporte

a todas las técnicas y ejemplos explicados, con el fin de facilitar una visita divulgativa a este interesante tema. Y parece ser que lo hemos conseguido, hasta el punto de que más de una vez nuestro editor, un reputado y experimentado divulgador matemático, nos ha preguntado: ¿pero este libro va realmente de matemáticas? Pues sí, va de matemáticas y, de hecho, sin las matemáticas todo lo explicado aquí no sería posible.

Polémica por las medidas del Gran Ordenador Central

Graves disturbios en las principales ciudades europeas en contra de los recortes sociales del Gobierno

París, Bruselas, Barcelona, Milán, Oporto, Munich y muchas otras ciudades europeas se han visto gravemente afectadas por las manifestaciones y revueltas populares en protesta por las últimas medidas aprobadas por el Gran Ordenador Central (GOC).

La nueva legislación tendrá un impacto directo en la gran clase media europea, ya que el paquete de medidas aprobado reduce en un 10% el número de días de vacaciones, pasando de 200 días anuales a 180, reduce la temperatura de los habitáculos en 1 °C, pasando de 25 °C a 24 °C, y se dejará de subvencionar el quinto robot asistente por ciudadano, implicando una reducción del 20% en el presupuesto asistencial.

Este paquete de medidas supone un ahorro de 8.000 billones de euros que serán destinados al aumento de la productividad minera en las colonias de Marte y Venus, según declaraciones del GOC, que desde hace 34 años gobierna con mano férrea el Sistema Europeo. Como es sabido, la Constitución Europea, revisada hace 39 años y ratificada por el Pueblo, otorga un gran poder ejecutivo, legislativo y judicial al GOC, cuya capacidad de computación, memoria y rapidez de análisis supera de largo la habilidad de cualquier equipo humano.

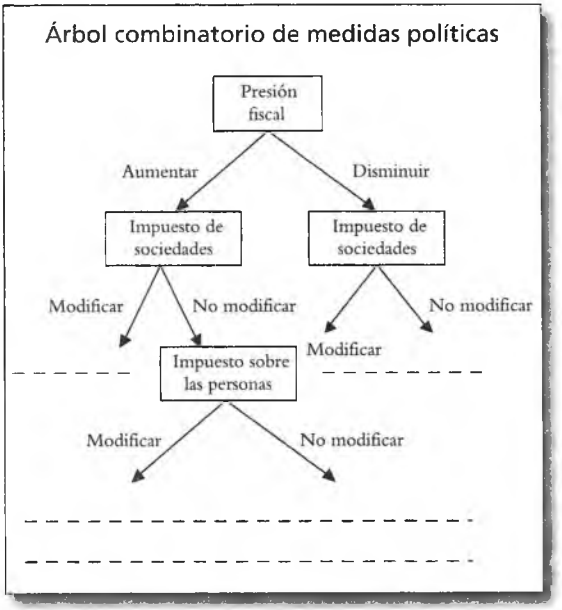
Aunque diversos Analistas Automáticos (AA) de los principales Sistemas Independientes de Análisis Automático (SIAA) han corroborado la efectividad del paquete de medidas del GOC, los representantes del Pueblo afirman que es un ataque frontal a las libertades ciudadanas en respuesta a la firme negativa humana a aprobar una Declaración Universal de Derechos de los Robots y Máquinas Autónomas.

En las entrañas del Gran Ordenador Central

El GOC está guiado por un «mapa» cognitivo con trillones de variables, cada una de ellas contenida en lo que que se conoce como «neuronas», las cuales, en el momento de la construcción, fueron interconectadas con las neuronas vecinas, creando así una gran red neuronal. Esta red neuronal es cambiante, es decir, dado un nuevo suceso, el valor de la variable que corresponde a dicho suceso cambia, y a continuación, como si de una reacción en cadena se tratara, lo hacen las neuronas conectadas a ella.

Por explicarlo gráficamente, es como si tiráramos una piedra sobre una piscina. El lugar donde la piedra ha impactado con la superficie del agua sufre un cambio en las tensiones superficiales, que son transmitidas casi instantáneamente a las otras moléculas de agua que hay en la superficie de la piscina, creando así las conocidas ondas que se van desplazando por toda la piscina hasta estabilizarse.

Cuando se construyó el GOC, se introdujeron en el mapa cognitivo cuatrillones de datos recogidos a lo largo de la historia de la humanidad. Seguidamente, el mapa cognitivo autoconstruyó sus conexiones neuronales en base a la experiencia pasada.



Una vez construida la red de conexiones neuronales, el GOC la usa no sólo para tener controlado el Sistema Europeo sino también para tomar decisiones de actuación, y esto lo hace a través de un sistema de razonamiento automático. Este sistema es capaz de lanzar hipótesis y predecir sus efectos. Siguiendo con el símil anterior, se lanza la piedra a la piscina, pero no en la piscina real, sino en una copia de la piscina original, y se observa cómo cambia la superficie del agua. Si el efecto es positivo, se ejecuta dicho cambio.

Simplificando mucho, el sistema de razonamiento automático del GOC es un algoritmo de búsqueda. Si el GOC detecta que debe solucionar un problema o alguna carencia, despliega un árbol de combinaciones que reflejan todas las medidas políticas, sociales, económicas e incluso militares que se podrían adoptar. Como la política es tan y tan compleja, este árbol, a los pocos milisegundos ya contiene millones de ramificaciones, y si no se hiciera nada más, a los pocos segundos tendría más ramificaciones que átomos hay en el Universo.

Supongamos que el objetivo es reducir el déficit fiscal en un 1,5%. Con la velocidad de cálculo de los supercomputadores cuánticos actuales, un árbol combinatorio como éste se convierte en prácticamente infinito a los pocos segundos de iniciarse el proceso. Como es obviamente imposible computar un árbol infinito, el GOC usa diversas herramientas para «podar» aquellas ramas del árbol por las que va viendo que no es viable conseguir el objetivo deseado. Por ejemplo, si se quiere reducir el déficit fiscal y no se modifica ningún impuesto ni se anima al crecimiento de la economía mediante una expansión fiscal, ya puede predecir que esta vía conduce a un camino sin salida. Por tanto, todas aquellas ramas del árbol que representen este tipo de soluciones ya se pueden descartar. Estas herramientas para la «solución de problemas», conocidas como *heurísticas*, se construyeron de forma automática a partir de datos históricos. Con posterioridad, un equipo de investigadores en ciencias sociales de toda Europa hizo una revisión pormenorizada de ellas. Para sorpresa de los escépticos, las variaciones introducidas por los expertos en las primeras versiones de las heurísticas sólo son un 0,003% con respecto al total. La revisión experta tardó 5 años en completarse, mientras que la deducción automática sólo había tardado 3 días.

Este sistema de deducción automático de heurísticas está basado en algoritmos evolutivos, es decir, en «sistemas inteligentes» que proponen soluciones aleatorias (en este caso, cada solución es una heurística propuesta), y se va

refinando a lo largo del tiempo, tratando de simular un proceso de evolución natural, completamente basado en las leyes de la evolución dictadas por Darwin. Es decir, las soluciones mejor adaptadas al medio son las que más descendientes dejan, o en otras palabras, las heurísticas que más respaldo tienen por parte de los datos históricos, más probabilidades tienen de propagarse en este proceso de evolución virtual y de dejar descendencia.

Ahora tan sólo queda esperar para comprobar que las medidas propuestas por el Hércules v3.4 llevan a nuestra sociedad a buen término, tal y como ha venido sucediendo en los últimos 34 años, y ver si al final todo ello redunda en un aumento sustancial de nuestra calidad de vida.

Como habrá adivinado el lector, esta noticia es totalmente ficticia y queda todavía muy lejos de lo que es posible en la actualidad. ¿Pero sería posible un escenario como éste dentro de 50 años? ¿Un escenario donde las grandes decisiones que dirijan el destino de la humanidad sean tomadas, controladas, vigiladas y analizadas por máquinas pensantes?

De hecho, como se verá en el capítulo 4, en Chipre, donde la situación política y militar es extremadamente compleja, investigadores de la universidad local y el Banco de Grecia ya han propuesto un sistema basado en mapas cognitivos para predecir la estabilidad del sistema ante cambios planteados por cualquiera de los actores implicados: griegos, turcos, OTAN, Unión Europea, etc.

Para hacernos una idea más aproximada de hasta qué punto este relato inicial es plausible o no, veamos en qué situación se encuentra actualmente la inteligencia artificial para, a partir de ahí, intentar averiguar cuán lejos se sitúa un escenario como éste. Bienvenidos al apasionante mundo de la inteligencia artificial, donde matemáticas, computación y filosofía se dan la mano y se asoman al límite mismo de lo que nos hace humanos.

Capítulo 1

¿Qué es la inteligencia artificial?

A menudo vemos en la televisión películas de ciencia ficción en las que las máquinas son autónomas y capaces de tomar decisiones por ellas mismas. ¿Qué hay de real y de ficción en todo ello? ¿Hasta qué punto está avanzada hoy en día la inteligencia artificial? ¿Llegará pronto el momento en que podamos desarrollar sistemas como los que aparecen en *2001: Una odisea en el espacio* o, más recientemente, en la versión cinematográfica de *Yo, robot*?

Antes de empezar, sin embargo, debemos dar una definición precisa del concepto que nos ocupa. Con «artificial» seguramente todos estaríamos de acuerdo: «no natural», es decir, «hecha, creada por la voluntad humana». Ahora bien, ¿qué es la «inteligencia»? Según la mayoría de diccionarios, la palabra «inteligencia» tiene numerosas acepciones, entre ellas las de «capacidad de entender o comprender», «capacidad de resolver problemas» o «habilidad, destreza y experiencia». En realidad, el hecho de que existan definiciones tan diferentes para este término pone de manifiesto la complejidad que hay detrás de tal concepto.

Los psicólogos y filósofos han tratado de delimitar, definir y medir la inteligencia a lo largo de los siglos. Sin embargo, estas métricas son todavía más confusas cuando se aplican a una entidad no humana. Por ejemplo, ¿diríamos que es inteligente un programa de ordenador que puede sincronizar y coordinar un complejo sistema motor de vuelo, que dispone de un sistema autónomo para decidir la ruta aérea en función de las necesidades del momento y que posee una fiabilidad del 100%? Probablemente sí. En cambio, ¿diríamos lo mismo de un mosquito? Este insecto es capaz de coordinar un complejo sistema motor de vuelo, toma decisiones autónomas de rutas aéreas y cuenta con una fiabilidad en sus operaciones aéreas del 100%.

El test de Turing

La primera solución práctica a la pregunta de si una determinada entidad no humana es inteligente o no llegó en el año 1950 de la mano del matemático Alan Turing,

considerado uno de los padres de la inteligencia artificial. Él es el autor del test de Turing, que determina la existencia de inteligencia en una máquina basándose en una idea muy simple: si una máquina se comporta en todos los aspectos como un ente inteligente, entonces debe de ser inteligente.

El test consiste en situar un evaluador humano y la máquina sujeta a evaluación en dos salas diferentes separadas por un tabique que impide que se vean. A continuación, mediante un teclado y una pantalla, el evaluador hace una serie de preguntas a la entidad que está siendo evaluada y ésta le responde. Si el evaluador considera que quien está respondiendo es un ser humano, se deduce que la máquina evaluada es inteligente y que, por tanto, tiene inteligencia artificial.

La persona que realiza las preguntas a la máquina (X) se las hace a un sistema con el que no tiene contacto visual. De esta manera, sólo a través del análisis de las respuestas puede evaluar si es o no un humano.

Como no podía ser de otro modo, el test de Turing recibió un aluvión de críticas por parte de algunos teóricos. ¿Se puede decir que una máquina es inteligente simplemente por responder a las preguntas basándose en un diccionario enorme de preguntas-respuestas? ¿O consiste la inteligencia en algo más que cierto comportamiento aparente, como, por ejemplo, la existencia de consciencia?

ALAN TURING (1912-1954)

Al matemático y filósofo inglés Alan Turing no sólo se lo considera uno de los padres de la inteligencia artificial, sino también de la informática moderna, por todos los avances teóricos fundamentales que aportó a esta ciencia en sus 42 años de vida. Durante la Segunda Guerra Mundial trabajó para el ejército británico en tareas de criptoanálisis, y fue uno de lo principales artífices de la ruptura de los códigos de la máquina Enigma, gracias a lo cual los aliados pudieron anticipar los movimientos del ejército nazi.

Su principal aportación teórica a la ciencia de la computación fue la hoy conocida como «máquina de Turing», un modelo teórico de computador universal. Un computador universal es aquel capaz de procesar cualquier dato de entrada y resolver su salida en un tiempo finito. La máquina de Turing consta de una cinta infinita con símbolos escritos sobre ella, un cabezal que puede desplazarse a izquierda o derecha sobre la cinta, leer los símbolos, borrarlos y escribir otros nuevos, y unas reglas que determinan el comportamiento del cabezal para cada posible símbolo que se detecte sobre dicha cinta. En la informática práctica estas reglas simbolizarían un programa informático, y la cinta representaría el sistema de entrada/salida del programa y un registro sobre su estado de ejecución.

Hoy en día, cuando se diseña un nuevo lenguaje de programación, como podría ser el C, el Pascal, el Java, etc., lo primero que se debe demostrar de manera formal es que el nuevo lenguaje es Turing-compatible, es decir, es equivalente a una máquina de Turing.

Por desgracia, Turing acabó su vida siendo perseguido por la justicia británica debido a su homosexualidad. En el transcurso de su juicio decidió no defenderse al considerar que no tenía de qué disculparse, por lo que fue considerado culpable, pero en la sentencia, Turing escogió la castración química antes que ir a prisión. Esto le provocó importantes alteraciones físicas que condujeron a su suicidio. En 2009 el primer ministro británico Gordon Brown pidió disculpas oficiales por el trato que recibió Turing durante sus últimos años de vida.

La principal crítica al test de Turing la escenificó el filósofo John Searle mediante el experimento de la habitación china. Imaginemos que unas personas cualesquiera que no saben ni una palabra de chino son introducidas en una habitación cerrada en un centro comercial de Shangai. A continuación se invita a las personas que pasan por allí a que formulen preguntas a los individuos que están dentro de la habitación; para ello deben introducirlas por escrito a través de una abertura. Por otro lado, dentro de la habitación hay un manual en el que están claramente representados todos los caracteres chinos que conforman las respuestas

¿SE PUEDE FINGIR LA INTELIGENCIA? AJEDREZ, KASPAROV Y DEEP BLUE

El ajedrez es un clásico problema combinatorio en el que, desde los principios de la informática, se han intentado aplicar, sin éxito, técnicas inteligentes para vencer a jugadores humanos. Pero ¿por qué no se puede fingir la inteligencia en juegos como el ajedrez? Imaginemos que introducimos en un ordenador las reglas del juego y éste construye el conjunto de todas las jugadas posibles. Seguidamente podríamos anotar, jugada a jugada, cuál es el mejor movimiento que conviene realizar ante cada una de ellas. Sin embargo, si calculamos el número de jugadas posibles, éste es del orden de 10^{123}. ¡Este número es mayor que el del total de electrones del universo! Por tanto, ¡sólo para almacenar los resultados deberíamos contar con una memoria mayor que la masa total del universo! Así pues, en el caso del ajedrez es totalmente imposible fingir la inteligencia mediante un diccionario de jugadas-movimientos, a diferencia de lo que sucedía en la habitación china.

El caso más polémico de programas de ordenador ajedrecistas fue el que enfrentó a la máquina Deep Blue y a Garry Kasparov. Deep Blue era un superordenador programado para jugar al ajedrez que venció por primera vez, en 1996, a un campeón del mundo humano. Sin embargo, del total de seis partidas que disputaron, el resultado final fue de 4-2 a favor del jugador ruso. En esa época, Deep Blue podía analizar 100 millones de movimientos por segundo. La polémica llegó cuando el campeón se enfrentó a una segunda versión, Deeper Blue, que podía analizar ya 200 millones de movimientos por segundo. En esa ocasión la máquina venció, pero Kasparov denunció que en un momento de la partida la máquina había sido ayudada por un operador humano. La situación fue la siguiente: él planteó una jugada donde, sacrificando un peón, podía armar un contraataque en jugadas subsiguientes. Esta trampa era imposible de detectar por el ordenador, ya que su capacidad de análisis era de un número limitado de jugadas futuras, en las que todavía no se desarrollaba el contraataque del jugador. Sin embargo, la máquina no cayó

a cada una de las posibles preguntas que se podrían formular. Por poner un ejemplo: un chico joven escribe en chino en una hoja de papel «¿Hace calor ahí dentro?» y la pasa por la rendija. Acto seguido, las personas que se encuentran en el interior de la habitación miran los caracteres chinos, los buscan en el manual y seleccionan una posible respuesta a esa pregunta. A continuación la transcriben, trazo a trazo, en una nueva hoja que pasan por la rendija de salida hacia el chico que hizo la pregunta. En la hoja de respuesta pone, en chino: «No, de hecho hace un frío que pela». Lógicamente, tanto ese joven chino como los demás evaluadores

en la trampa, lo cual levantó las sospechas del campeón ruso. Kasparov pidió posteriormente los registros por escrito de los procesos de la máquina, a lo cual IBM accedió, aunque al final nunca los presentó.

Deep Blue, el supercomputador de IBM que venció al campeón del mundo de ajedrez.

están recibiendo respuestas coherentes en su idioma, por lo que pensarán que las personas que hay dentro de la habitación saben hablar chino perfectamente. Sin embargo, los individuos de la habitación han basado sus respuestas en un manual de correspondencias y no han entendido ni una sola palabra de ninguna de las conversaciones.

Pero ¿podemos sospechar que una máquina que supera el test de Turing puede estar engañándonos de una manera similar a como lo hace la habitación china? La respuesta es que no. La habitación china es un escenario engañoso porque en realidad, si bien es cierto que las personas que había dentro de ella no sabían chino, quien realmente contestaba a las preguntas era un conjunto de entidades formado por las personas y el manual. Aunque no se puede decir que el manual «supiera» chino, en todo caso sí fue confeccionado por alguien que, efectivamente, conocía dicho idioma, ya que, si no, no habría sido capaz de escribir el conjunto de preguntas y respuestas.

De manera práctica, hoy en día se considera que una nueva tecnología es inteligente si es capaz de resolver un problema de una manera creativa, algo que desde siempre se ha considerado potestad exclusiva del cerebro humano. Un ejemplo representativo de tecnología que parece inteligente pero que no se la considera como tal es la de los primeros sistemas expertos que aparecieron en la década de 1960. Un «sistema experto» es un programa informático que se ha implementado con unas determinadas reglas, más o menos complejas, y que puede actuar de manera autónoma en el control de ciertos sistemas. Un ejemplo de ello podría ser un programa de ordenador con una enorme lista de síntomas médicos programados de manera que, dado un nuevo paciente y sus síntomas, puede determinar el tratamiento que dicho paciente necesita. Sin embargo, si el sistema no es capaz de crear una nueva regla deducida de las anteriores ni inventar un nuevo tratamiento cuando la situación lo requiera, se considera que no es creativo y, por ende, no es inteligente.

Así pues, para considerar que un programa informático es inteligente se deben cumplir unas condiciones un tanto subjetivas, como, por ejemplo, que tenga capacidad de aprender en materias complejas, de optimizar funciones matemáticas con muchos parámetros (dimensiones) y en un intervalo (dominio) enorme, o de planificar una gran cantidad de recursos con restricciones.

Al igual que sucede con el resto de las áreas de la ciencia y de la tecnología, la inteligencia artificial se ha especializado y dividido en cinco grandes ramas, que son las siguientes:

1. La búsqueda.
2. El aprendizaje.
3. La planificación
4. El razonamiento automático.
5. El procesamiento del lenguaje natural.

Sin embargo, las tecnologías y los algoritmos usados en las distintas especialidades muchas veces son los mismos. A continuación repasaremos estas ramas y daremos algunos ejemplos prácticos.

La búsqueda

La búsqueda hace referencia al proceso de buscar la solución óptima a un determinado problema. Cuando dicho problema es definible mediante una función matemática estaríamos hablando de optimización de funciones, es decir, de la búsqueda de los parámetros de entrada que maximizan la salida de la función. A menudo hay problemas que implican la optimización de varias funciones a la vez, y estas funciones son, además, difíciles de definir y de delimitar. Para un sistema automático, la optimización de funciones es un problema complejo, en especial si no se dispone de una fórmula analítica de la función y sólo se puede inferir la «forma» de la función a partir de unas pocas muestras. Además, a menudo sucede que la función en cuestión tiene centenares de parámetros distintos que ajustar, o que para la obtención de cada una de las muestras se necesitan varias horas de cálculo, o incluso que las muestras contienen ruido, es decir, que el valor de la función en un determinado punto del espacio no es exacto.

Para abordar estos escenarios complejos se utiliza la inteligencia artificial. Nótese que el ser humano puede resolver en instantes complejas funciones matemáticas multidimensionales de manera instintiva. Un caso ejemplar son las funciones de similitud. Imagínese que usted conoce a más de 500 personas, pero si observara la fotografía de un determinado individuo, rápidamente podría decir si esa fotografía corresponde a alguno de sus conocidos y a cuál de ellos. Esta, en apariencia, sencilla operación se resuelve mentalmente mediante la optimización de una función que mide las diferencias entre los rostros que usted tiene memorizados y el rostro fotografiado que está observando. En un rostro hay miles de características o dimensiones observables, como, por ejemplo, el color de los ojos, la relación de tamaño entre la boca y la nariz, la existencia de pecas, etc. Nuestro cerebro es capaz de detectar

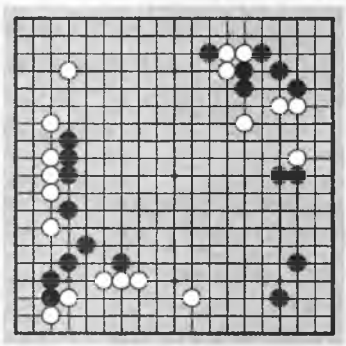
EL GO, UNO DE LOS GRANDES RETOS PENDIENTES DE LA INTELIGENCIA ARTIFICIAL

El go es un buen ejemplo de problema combinatorio donde un humano medianamente entrenado puede detectar de un vistazo la estrategia más inteligente para cada escenario, pero que, en cambio, resulta extremadamente complejo para un ordenador. Hasta el momento, ningún programa de ordenador ha sido capaz de vencer a ningún jugador profesional sin desventajas iniciales.

Este juego de estrategia de origen chino tiene unas reglas extremadamente sencillas que dan lugar a escenarios de gran complejidad estratégica. Se trata de un tablero con una cuadrícula de 19 × 19 líneas donde dos jugadores van ubicando, en turnos alternativos, fichas blancas y negras en las intersecciones libres. Si una ficha o grupo de fichas queda completamente rodeado por las fichas del color contrario, el grupo queda capturado y estas fichas se retiran del tablero. Se puede pasar el turno si se cree conveniente, pero si los dos jugadores pasan consecutivamente la partida se acaba y gana aquel jugador que en ese momento esté dominando una mayor porción del tablero de juego.

Matemáticamente, el go se clasifica como un juego de estrategia en todo similar al ajedrez. Sin embargo, mientras que existen programas informáticos capaces de vencer a los campeones mundiales de ajedrez, es difícil que un programa de go pueda ganar a un jugador aficionado. Esto sucede principalmente por tres razones: primero, por las dimensiones del tablero de go, que es más de cinco veces mayor que el de ajedrez, lo cual implica un mayor número de jugadas que analizar; segundo, porque un movimiento de go puede afectar en centenares de turnos posteriores, de modo que es casi imposible para un ordenador hacer predicciones a tan largo plazo, y, finalmente, porque en ajedrez las piezas se capturan de una en una y todas tienen un determinado valor, por lo que se puede evaluar con bastante precisión qué beneficios dará una jugada, mientras que en el go, en cambio, el beneficio que se obtenga de una captura depende por completo de las piezas que se capturen, lo cual viene dado por la situación de las piezas en aquel momento.

todas esas características y compararlas con las de los rostros de todas las personas que conocemos; puede medir las distancias correspondientes al rostro de la fotografía y compararlas con las de todos los demás, y así encontrar el rostro en el que la distancia es mínima. Además, es capaz de decidir si esa distancia es lo suficientemente reducida como para deducir que la persona fotografiada y la recordada son o no la misma. El cerebro humano realiza todas estas operaciones en menos de un segundo. Sin embargo, para un ordenador el reconocimiento de rostros es una ope-

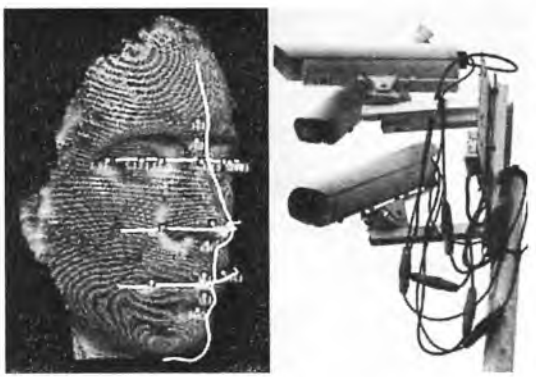


Arriba, tablero y fichas de go; estas últimas reciben el nombre de «piedras». A la izquierda, situación de la partida en la final del campeonato mundial celebrado en el año 2002 entre Choe Myeong-hun (blancas) y Lee Sedol al final de la apertura.

ración extremadamente compleja y, probablemente, un moderno computador tardaría minutos en obtener la solución.

Pero ¿por qué llamamos «búsqueda» a esta rama de la inteligencia artificial si hemos estado hablando de optimización numérica? La búsqueda incluye la solución a otro tipo de problemas, por ejemplo, los llamados «problemas combinatorios». Un problema combinatorio es aquel cuya solución está formada por distintos elementos que pueden combinarse entre sí y dar lugar a un espacio combinatorio.

La solución viene determinada por el conjunto de elementos óptimo. Un buen ejemplo combinatorio sería una partida de ajedrez, en la que la solución óptima termina siendo una sucesión de movimientos de las piezas que llevan a ganar la partida. Otro ejemplo clásico es el conocido como problema de la mochila, en el que existen diversos objetos que podemos meter en una mochila antes de una excursión. En este caso la solución es la combinación de objetos que minimiza el peso de la mochila pero maximiza el valor de los objetos contenidos en ella. De nuevo, a menudo sucede que un problema combinatorio relativamente sencillo para un humano es extremadamente complejo de resolver para una computadora.



Éste es uno de los muchos sistemas informáticos utilizados para el reconocimiento de formas, en este caso de rostros. La presente imagen responde a un desarrollo de la compañía japonesa NEC.

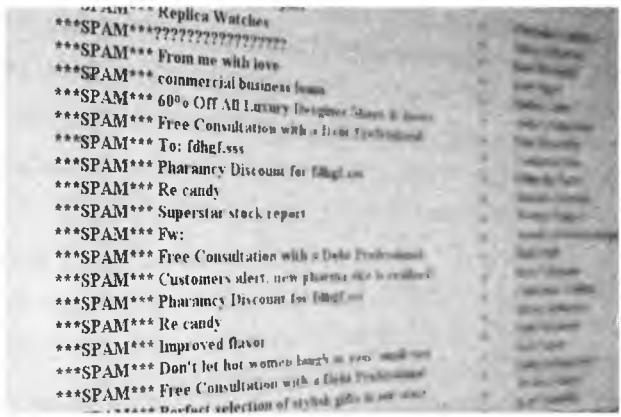
El aprendizaje

La siguiente rama de la inteligencia artificial es el aprendizaje. ¿Es inteligente un sistema que puede aprender de experiencias pasadas? Recuperemos el ejemplo del sistema de diagnóstico médico automático, al que se le ha introducido un conjunto de síntomas asociados a una patología. A este proceso de introducción de información asociada entre sí se le llama entrenamiento. Así pues, una vez entrenado el sistema, cuando se le da un nuevo síntoma es capaz de rebuscar en su memoria para ver si éste apareció en el pasado y, de ser así, responder con la patología que tenía asociada. En este caso, se dice que el sistema aprende a base de memorización y que, por tanto, no es inteligente. La rama del aprendizaje se basa en entrenar

sistemas para que después sean capaces de generalizar, es decir, de deducir unas reglas que luego puedan aplicarse a nuevos problemas que no han aparecido anteriormente.

El aprendizaje automático ha sido una de las áreas más prolíficas de la inteligencia artificial. Numerosas universidades, centros de investigación y empresas aportan a diario avances en este campo. Probablemente esto se debe, por un lado, a la gran necesidad de sistemas expertos que existe en determinadas áreas del conocimiento y en la industria, y, por otro, a la complejidad de la programación de los sistemas expertos útiles. A un sistema experto inteligente se lo entrena con una serie de casos asociados a sus soluciones para que sea capaz de deducir las reglas y normas que los asocian. Después, dado un nuevo caso, el sistema puede determinar la nueva solución. Así, para que un sistema experto se considere inteligente y sea útil, es fundamental el hecho de que pueda aprender y generalizar de manera automática, es decir, que no se le tengan que introducir las reglas manualmente, y que, una vez entrenado, pueda comportarse como un experto en la materia para la que ha sido entrenado.

Más adelante desarrollaremos con detalle el tema de las aplicaciones de los sistemas expertos, aunque cabe citar algunos ejemplos ilustrativos de sistemas expertos actuales, como, por ejemplo, los de predicción de morosidad en hipotecas bancarias, de detección precoz de tumores malignos o de clasificación automática de correo electrónico no deseado (*spam*).



La clasificación automática del correo electrónico, con el fin de separar el spam *del correo legítimo, es una de las aplicaciones de los sistemas expertos.*

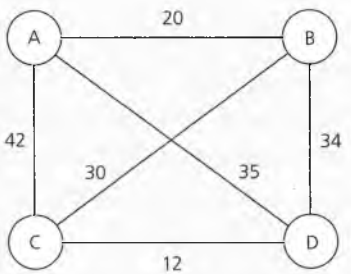
La planificación

La tercera gran rama de la inteligencia artificial es la planificación. Esta capacidad ha acompañado al ser humano desde tiempos inmemoriales y, de hecho, es la que ha permitido su supervivencia a lo largo de la historia. Si nos remontamos al Paleolítico, nos podemos encontrar ya con un dilema que requiere planificación: dados unos recursos alimenticios concretos y una serie de consumidores, que son los integrantes de la tribu, ¿cómo se tiene que distribuir el alimento entre todos los

EL PROBLEMA DEL VIAJANTE

A menudo, un determinado problema puede ser clasificado en una u otra rama de la inteligencia artificial según cómo se enfoque su resolución. Un buen ejemplo es el famoso problema del viajante (también conocido como TSP por sus iniciales del inglés: *Travelling Salesman Problem*), el cual se puede resolver siguiendo una estrategia de búsqueda o bien de planificación.

El enunciado dice así: dados un conjunto de ciudades, las rutas que las interconectan y la distancia que las separa, planifíquese la ruta que debería realizar un representante comercial que tiene que visitar clientes en cada una de las ciudades. Se pide que el viajante no repita ciudad y que, además, el número final de kilómetros recorridos sea el mínimo. Como el lector puede intuir, en algún caso es posible que, en función del aspecto de las rutas entre las ciudades, sea estrictamente necesario repetir una ciudad para poder visitarlas todas; por tanto, la repetición de una ciudad se puede ver como una condición violable.



Ejemplo de grafo de ciudades conectadas entre sí a la distancia en kilómetros que indican los números de las aristas.

individuos? ¿Daremos las carnes más suculentas y más ricas en recursos energéticos al personal encargado de la recolección de bayas o a los fuertes cazadores de la tribu? Pero ¿qué pasa si una de las recolectoras está en avanzado estado de gestación? Bien, todas estas preguntas corresponden a lo que llamamos las «restricciones del sistema», es decir, aquellas circunstancias que deben tenerse en cuenta cuando se elabora un plan.

Las restricciones básicamente pueden ser de dos tipos: violables e inviolables. En el ejemplo anterior de la tribu prehistórica, aunque las mejores piezas de carne deberán ir a quien más las necesita, no pasa nada si un día el cazador más fuerte de la tribu no se lleva el trozo más suculento de la presa del día. Aunque esta situación es insostenible en el tiempo, el individuo puede aguantar su hambre por un día. Por tanto, ésta es una restricción violable.

El caso contrario podría ser, por ejemplo, la distribución de los recursos de una gran universidad (aulas y profesores) para planificar un año escolar. En este caso los consumidores de recursos serían el conjunto de estudiantes matriculados en la asignatura de, por ejemplo, cálculo numérico, el de los matriculados en derecho mercantil, el de los matriculados en física, etc. A la hora de hacer la distribución, deberemos tener en cuenta que el grupo de derecho mercantil y el de física no pueden estar compartiendo el aula 455 al mismo tiempo. El catedrático de cálculo numérico tampoco puede estar dando una clase de derecho mercantil en ningún momento del año, ya que, probablemente, no está cualificado para ello. En este ejemplo, las restricciones son inviolables.

La violabilidad o inviolabilidad de los recursos es crítica y constituye un aspecto fundamental que debe tenerse en cuenta cuando se programa un algoritmo inteligente de planificación.

El razonamiento automático

La cuarta rama de la inteligencia artificial es el razonamiento automático. Sin duda alguna, ésta es la rama que más interés y fascinación ha despertado entre el gran público y, de hecho, constituye un tema habitual de las películas y de los libros de ciencia ficción. Sin embargo, este campo nació en el entorno, ya no tan glamuroso, de la demostración automática de teoremas matemáticos.

A menudo se plantean nuevos teoremas que los matemáticos deben demostrar si son ciertos o no, proceso que puede revestir una gran complejidad. Es lo que ocurrió con el teorema de Fermat (en el que si n es un número entero mayor que

dos, entonces no existen números naturales no nulos que cumplan la igualdad $z^n = x^n + y^n$), que ¡tardó en ser demostrado más de 200 años!

En este contexto, en el año 1956, el economista Herbert Simon (1916-2001) y el ingeniero Alan Newell (1927-1992) desarrollaron conjuntamente la máquina Logic Theorist, capaz de demostrar teoremas nada triviales de lógica matemática. Desde luego, el desarrollo de esta máquina marcó un importante hito en la disciplina de la inteligencia artificial y reavivó las discusiones filosóficas sobre la posibilidad de construir máquinas pensantes; sin lugar a dudas, muchos de los libros y de las películas de las décadas de 1960 y 1970 en los que aparecían malévolas máquinas inteligentes estuvieron influidos por estas discusiones. Según la influyente filósofa Pamela McCorduck, Logic Theorist es la prueba de que una máquina puede ejecutar tareas consideradas inteligentes, creativas y únicamente realizables por un humano.

Herbert Simon (izquierda) y Allen Newell jugando al ajedrez en 1958.

Logic Theorist usaba lo que se conoce como «sistemas simbólicos», unos sistemas inventados por los matemáticos para dar sentido a algunas expresiones sin referirse a convenciones arbitrarias. Por ejemplo, podemos decir que «ser un hombre» implica «ser mortal», sentencia que puede formalizarse mediante la expresión matemática «A → B», en la que el símbolo «A» equivale a «ser un hombre», el

símbolo «→» significa «implica» y «B» equivale a «ser mortal». «Ser un hombre implica ser mortal» es una expresión arbitraria que se formaliza mediante la expresión «A → B». Una vez que los términos arbitrarios están formalizados, es mucho más sencillo manipularlos y hacer operaciones con ellos desde un punto de vista informático o matemático.

Con el propósito de simplificar las operaciones matemáticas, los sistemas simbólicos parten de axiomas para construir teoremas con ayuda de reglas de derivación. La ventaja de los sistemas simbólicos es que, al ser sistemas formales perfectamente definidos y sin ambigüedades, su programación informática es relativamente sencilla. Veamos un ejemplo:

Sócrates es un hombre.
Todos los hombres son mortales.
Por tanto, como Sócrates es un hombre, es mortal.

Si lo formalizamos matemáticamente, estas sentencias quedarían de la siguiente manera:

A: Sócrates
B: hombre(s)
A → B
C: mortal(es)
B → C
Si A → B y B → C, entonces A → C, es decir, Sócrates es mortal.

En este caso, una regla de derivación conocida como «silogismo hipotético» nos permite concluir que A → C si es cierto que A → B y B → C.

Sin embargo, la derivación automática y sistemática de teoremas a partir de los axiomas y las reglas de derivación puede conducirnos a un número de combinaciones que, de nuevo, nos acerca peligrosamente al número de átomos del universo. Por ello, Logic Theorist se valía de consideraciones heurísticas, es decir, de un instrumento vagamente predictivo que ayuda a seleccionar las mejores derivaciones de entre todas las posibles, para identificar la secuencia correcta de derivaciones que se deben realizar sobre los axiomas hasta llegar a la demostración de los teoremas.

A continuación mostramos un ejemplo práctico. Queremos saber si Sócrates es mortal o no lo es y conocemos los siguientes axiomas iniciales:

A: Sócrates
B: hincha del Olympiacos
C: griego
D: hombre
E: mortal

$A \rightarrow C$
$C \rightarrow D$
$A \rightarrow D$
$C \rightarrow B$
$D \rightarrow E$

Y queremos saber si $A \rightarrow E$ es verdadero o falso mediante «fuerza» bruta, es decir, probando todas las combinaciones posibles; de este modo tenemos:

$A \rightarrow C \rightarrow D \rightarrow E$
$A \rightarrow C \rightarrow B$
$A \rightarrow D \rightarrow E$

Es decir, hemos realizado siete operaciones lógicas partiendo sólo de cino axiomas y usando únicamente una regla de derivación: el silogismo hipotético. Como el lector podrá imaginarse, en escenarios mas complejos, con más axiomas y más reglas de derivación en uso, el número de combinaciones posible puede ser tan elevado que se tardarían años en obtener demostraciones concluyentes. Para atajar este problema, tal y como propusieron Simon y Newell, una consideración heurística (o como dicen los especialistas, «una heurística») nos habría avisado en el ejemplo anterior de que no vamos por el buen camino si para demostrar que alguien es mortal tenemos que empezar a hablar de fútbol ($A \rightarrow C \rightarrow B$).

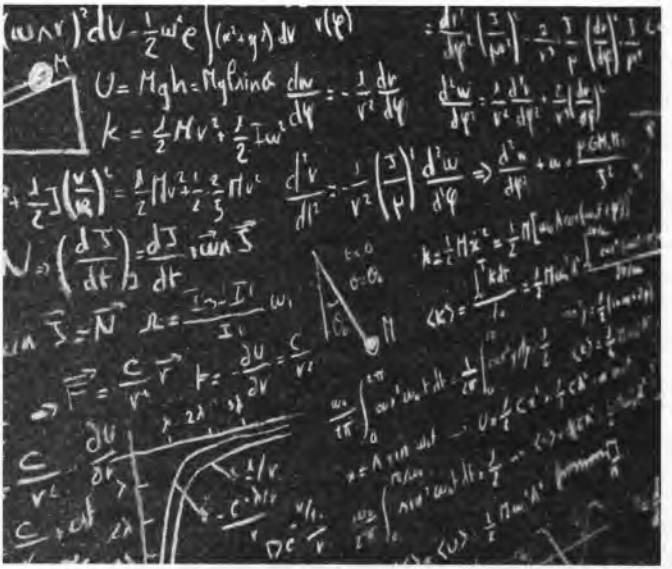
En la actualidad, más allá de los sistemas automáticos de demostración de teoremas matemáticos, tanto los sistemas simbólicos como las heurísticas son ampliamente utilizadas en la resolución de problemas prácticos.

Para ilustrar otro caso del uso de heurísticas volveremos al campo del ajedrez. En un turno de ajedrez existen de media 37 posibles movimientos. Por ello, si un programa de ordenador pretendiera analizar una jugada con 8 turnos de profundidad tendría que analizar el equivalente a 37^8 escenarios posibles, es decir 3.512.479.453.921 jugadas, o, en otras palabras, más de 3,5 billones de jugadas. Si el ordenador invirtie-

LÓGICA MATEMÁTICA

La lógica matemática es una parte de las matemáticas que se encarga de estudiar la forma del razonamiento, es decir, es una disciplina que, por medio de reglas y técnicas, determina si un argumento es válido o no. La lógica es ampliamente utilizada en filosofía, matemáticas y, por supuesto, computación como instrumento para validar o derivar nuevo conocimiento. Fueron George Boole, con su álgebra denominada luego «booleana», y Augustus De Morgan, con sus leyes lógicas, quienes desarrollaron las bases lógicas aristotélicas y, mediante una nueva notación más abstracta, desarrollaron este instrumento útil para investigar sobre los fundamentos de la matemática.

En los últimos cincuenta años, la lógica matemática ha experimentado grandes avances y ha dado lugar a la llamada «lógica moderna». Para distinguirla de la lógica clásica, a esta última se la denominó «lógica de primer orden». Formalmente, la lógica de primer orden implica sólo expresiones finitas y fórmulas bien definidas, sin lugar para dominios infinitos ni para la incertidumbre.



Por complicada que parezca una expresión escrita en la pizarra, muy raramente utilizará un marco de significado fuera de la lógica de primer orden.

ra un microsegundo en estudiar cada jugada, esto querría decir que sólo para analizar 8 niveles del juego (algo que para un jugador experto es bastante sencillo) ¡un ordenador potente estaría pensando más de dos años y medio por turno!

Así pues, ya podemos intuir que es necesario introducir algún tipo de mejora en el método que permita acelerar el proceso, y esa mejora son las heurísticas. Se trata de unas reglas predictivas que ayudan al algoritmo a descartar aquellas jugadas que, por alguna razón, se percibe que van a conducir a situaciones muy desfavorables, de modo que no hace falta continuar explorándolas. Sólo con que las heurísticas eliminen el análisis de unas pocas jugadas absurdas, el ahorro en número de jugadas que se deben analizar puede ser enorme. En resumen, las heurísticas son herramientas predictivas fuertemente basadas en la intuición del programador y resultan tan

GEORGE BOOLE (1815-1864) Y SU LÓGICA

Si decimos que Alan Turing es uno de los padres de la informática moderna, de George Boole se puede decir que es el abuelo. En efecto, este matemático y filósofo británico desarrolló el álgebra de Boole, la base de la aritmética computacional moderna, en la cual se fundamenta toda la electrónica digital.

Boole ideó un sistema de reglas que, mediante procedimientos matemáticos, permiten expresar, manipular y simplificar problemas lógicos que admiten dos estados, verdadero o falso. Las tres operaciones matemáticas básicas del álgebra booleana son la negación o complemento, la unión u «o» y la intersección o «y». La negación, representada con el símbolo ¬, consiste en invertir el estado de una variable. Por ejemplo, si A = «Aristóteles es un hombre», entonces ¬A = «Aristóteles no es un hombre». La unión, representada con el símbolo v, es un operador binario, es decir, que necesita dos parámetros para obtener el resultado. Éste es verdadero si alguno de los dos

fundamentales en la mayoría de sistemas inteligentes que condicionan en gran medida la calidad de los mismos.

En los últimos años, la rama del razonamiento automático ha ido avanzando progresivamente hasta llegar a poder razonar en sistemas incompletos, inciertos y no-monótonos, es decir, en entornos en los que falta información (incompletitud), puede haber afirmaciones iniciales contradictorias (incertidumbre) o, al introducir nuevos conocimientos al sistema, el conocimiento global sobre el entorno no tiene por qué aumentar (no-monotonía).

Una herramienta de enorme potencia para trabajar en estos entornos es la lógica difusa, esto es, un tipo de lógica matemática en la que las afirmaciones no tienen por qué ser todas ciertas o falsas. Mientras que en la lógica booleana «clásica» dada

parámetros es verdadero. Por ejemplo: «¿Es verdadero que lo que usted está haciendo ahora es leer "o" conducir?». La respuesta es «Sí, es verdadero», dado que lo que usted está haciendo ahora es leer este libro. Si ahora usted estuviera conduciendo y no leyendo, la respuesta a la pregunta también sería afirmativa. Incluso sería igualmente verdadera si estuviera cometiendo la grave imprudencia de conducir y leer este libro al mismo tiempo. Finalmente, el tercer operador es la intersección, representada con el símbolo $\wedge$ y que también es un operador binario. Si reformulamos la pregunta anterior y decimos «¿Es verdadero que lo que usted está haciendo ahora es leer "y" conducir?», sólo podríamos contestar «Sí, es verdadero» si estuviéramos cometiendo tamaño disparate.

A partir de estos tres operadores, se pueden construir otros operadores más sofisticados, como la o-exclusiva ($\overline{\vee}$), que sólo sería cierta, en el segundo ejemplo, si estamos leyendo o conduciendo, pero no si hacemos las dos cosas a la vez. El operador $\overline{\vee}$ no es un operador básico del álgebra booleana dado que se puede reescribir mediante los otros tres operadores básicos: $A \,\overline{\vee}\, B = (A \wedge \neg B) \vee (\neg A \wedge B)$.

En electrónica digital «verdadero» y «falso» se simbolizan con un 1 y un 0, respectivamente, y se les da el significado de dejar pasar la corriente eléctrica (1) o no dejarla pasar (0). Las operaciones lógicas se realizan mediante la combinación de transistores y ha sido la alta integración de estos dispositivos electrónicos lo que ha permitido que la informática haya avanzado tanto en los últimos 40 años. El microprocesador del ordenador de su casa, que es el cerebro central del aparato, ¡contiene centenares de millones de transistores distribuidos en unos pocos centímetros cuadrados!

una afirmación siempre se puede decir si ésta es cierta o falsa (por ejemplo, es falso que «algún hombre es inmortal» y es cierto que «todos los hombres son mortales»), en cambio en la lógica difusa se incorporan gradaciones entre la verdad y la falsedad. Así, antes, si se decía que Creso no era pobre, significaba automáticamente que era rico, y si se afirmaba que Diógenes no era rico, esto quería decir que era pobre (en este ejemplo, la lógica booleana ¡discrimina a la clase media!); pero ahora, usando la lógica difusa, podríamos decir que Aristóteles fue rico en un grado de, por ejemplo, 0,6.

El procesamiento del lenguaje natural

Entender el lenguaje natural de un humano, así como poder comunicarse mediante él, es uno de los grandes retos pendientes de esta área de la ciencia. Que un sistema artificial pueda entender a un humano en su lenguaje natural, independientemente del idioma que éste utilice, sin necesidad de los lenguajes de programación, ayudaría en gran medida a humanizar las máquinas, al margen de su inteligencia. Ahora bien, por poder «procesar el lenguaje natural» no se entiende solamente el hecho de transliterar sonidos en fonemas y en signos escritos, sino también entender los conceptos trasmitidos (ya sea de forma escrita u oral) para llevar a cabo alguna acción con ellos.

Los primeros avances en el procesamiento del lenguaje natural vinieron de la mano del ingeniero Joseph Weizenbaum (1923-2008) entre 1964 y 1966 con su programa Eliza. A este programa informático, que simulaba ser un psicoterapeuta, los usuarios le podían hacer preguntas o comentarios, y Eliza parecía que los entendía, pues a continuación era capaz de responder, más o menos, con sentido. Una conversación real que mantuvo el autor de este libro con Eliza mientras lo escribía es:

I. Belda: Necesito unas vacaciones.
Eliza: ¿Que significaría para ti si tuvieras vacaciones?
I. Belda: Pues podría estar más tiempo con mi esposa.
Eliza: Cuéntame más sobre tu familia.
I. Belda: Bien, sólo tengo una esposa.
Eliza: ¿Dices que tienes sólo una esposa?
I. Belda: Sí, no soy polígamo.
Eliza: Pareces ser muy positivo.
I. Belda: Ella se llama Laia.

Eliza: ¿Y quién más hay en tu familia?
I. Belda: Nadie más.
Eliza: De verdad, ¿nadie más?
I. Belda: Pues no.
Eliza: ¿Estás seguro?
I. Belda: Segurísimo.

Como el lector puede apreciar, por suerte para los psicólogos, Eliza no les quitará el trabajo de manera inminente. Sin embargo, desde la década de 1960 el procesamiento del lenguaje natural ha ido avanzando de manera significativa y hoy en día existen técnicas de procesamiento del lenguaje natural que se utilizan para aspectos tan diversos como la detección de plagios literarios, la traducción automática de textos o la construcción de interfaces hombre-máquina más humanizadas. Si el lector tiene curiosidad, puede conversar online con Eliza en: http://www.chayden.net/eliza/Eliza.html.

Para terminar, gestionar lo que se sabe

Estructurar adecuadamente el conocimiento es muy importante, y para ilustrar el porqué de esta importancia pondremos un ejemplo: imaginemos que alguien nos pregunta quién es el alcalde de Austin (Texas). Probablemente, si no somos de esa zona de Estados Unidos, responderemos rápidamente: «No lo sé». En cambio, un sistema que no tenga bien estructurado su conocimiento, como puede ser cualquiera de nuestros ordenadores personales, invertirá unos cuantos minutos en analizar todos los documentos de su disco duro buscando si en algún documento se explicita el nombre del alcalde de dicha población. Un sistema inteligente, o que pretenda serlo, debería responder de una manera tan tajante como lo haría un humano y, para ello, el conocimiento debe estar bien estructurado y ser fácilmente accesible.

En la resolución de un problema práctico no sólo se debe estructurar adecuadamente el conocimiento, sino que además se tienen que programar las herramientas adecuadas para navegar por él y mantenerlo ordenado. En esta base de conocimiento es donde el sistema aplica sus estrategias de razonamiento, búsqueda, aprendizaje, etc. y, por tanto, la base de conocimientos de un sistema inteligente es cambiante. Por ese motivo, los sistemas inteligentes necesitan motores de control del conocimiento que resuelvan, por ejemplo, las contradicciones que puedan ir apareciendo, que eliminen redundancias e incluso que generalicen conceptos.

Para tener un buen control sobre el conocimiento contenido en una base de conocimiento se necesita metainformación que explique cómo está representado internamente el conocimiento y que nos ayude a delimitarlo. Saber cómo está representado el conocimiento no es un asunto baladí, ya que éste puede ser estructurado en un sinfín de formas. Por eso, disponer de información acerca de la estructuración del conocimiento almacenado puede ayudar en gran manera a los sistemas automatizados a navegar por él.

Otro aspecto que debe tenerse en cuenta es la delimitación del conocimiento, puesto que saber qué abarca y hasta dónde llega nuestra base de conocimiento almacenado, de nuevo, ayuda al sistema informatizado a recorrerlo. De hecho, un humano puede trabajar fácilmente con la idea de la incompletitud de su conocimiento, pero a un sistema informatizado se le debe especificar muy bien qué es lo que sabe y lo que no. Por ello, uno de los primeros métodos para gestionar las bases de conocimiento fue la asunción de mundo cerrado (CWA, *Closed World Assumption*). La CWA fue propuesta por Raymond Reiter en 1978 y se basa en una afirmación sencilla pero de notables consecuencias: «Los únicos objetos que pueden satisfacer el predicado P son aquellos que lo deben hacer», o, en otras palabras, que todo aquel conocimiento que no se tenga registrado no es cierto.

Un buen ejemplo de ello sería: si nos preguntaran si una determinada persona trabaja en una empresa, para averiguarlo consultaríamos la lista del personal de la misma, y si la persona no figurara en ella diríamos que no trabaja allí.

La CWA fue un gran avance en su momento y facilitó muchísimo la gestión de bases de conocimiento. Sin embargo, como el lector habrá intuido, la CWA tiene importantes limitaciones, ya que un factor desconocido no implica, en la vida real, que sea automáticamente falso. Retomando el ejemplo anterior de la lista de personal de una empresa, ¿qué pasa si una persona que trabaja en ella no figura en la lista de trabajadores por el simple motivo de que la lista contiene un error o bien no está actualizada? Justamente éste es uno de los puntos más débiles de la CWA, la incorrección o inconsistencia de los datos de la vida real. Su otro punto débil es que nos obliga a utilizar razonamientos puramente sintéticos. Veámoslo con un ejemplo práctico:

Imaginemos que tenemos la siguiente lista de solteros y no solteros:
Soltero Juan
Soltero María
No-soltero David

Si alguien pregunta al sistema si Jorge es soltero, como esto es una lista de solteros, contestará que no, dado que no aparece en ella. Sin embargo, podemos construir una nueva lista de casados, que quedaría así:

No-casado Juan
No-casado María
Casado David

Y si ahora le preguntamos al sistema si Jorge está casado, también nos responderá que no. Con lo cual finalmente tendríamos que, por no disponer de información sobre el estado civil de Jorge, el sistema llega a la conclusión incongruente de que no está casado ni soltero. Podemos ver claramente que el CWA no funciona bien en casos de incertidumbre o incompletitud del conocimiento y, por ello, hoy en día sólo se usa en la resolución de problemas muy particulares.

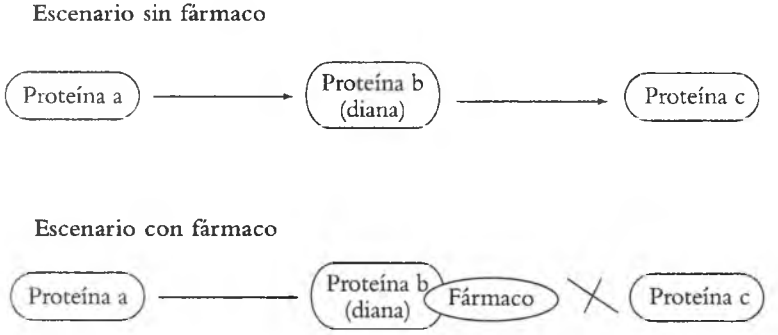
Por último, no podemos terminar con la gestión de bases de conocimiento sin hablar de los sistemas de mantenimiento de la verdad (TMS, *Truth Maintenance Systems*). Los TMS son los elementos que vigilan y controlan que la base de conocimiento sea consistente con ella misma, y resultan especialmente útiles cuando se usan métodos de razonamiento no-monótonos, es decir, aquellos en los que la base de conocimiento va aumentando o disminuyendo a medida que se razona. Los TMS pueden ser de dos tipos: de «búsqueda vertical» o de «búsqueda horizontal». Los primeros recorren la base de conocimiento yendo de lo general a lo particular en busca de contradicciones y, en caso de que detecten una, deshacen el camino recorrido para solventarla. Por el contrario, los de búsqueda horizontal plantean diferentes escenarios o hipótesis paralelas, de manera que el universo de contextos se va podando a medida que se detectan contradicciones. Es decir, dado un contexto posible (piénsese en una determinada situación de las piezas del ajedrez), identifica los diferentes escenarios en los que puede evolucionar la situación actual (en el caso del ajedrez, serían los posibles movimientos), y elimina aquellos que resultan contradictorios (en ajedrez un escenario contradictorio sería una jugada muy desfavorable para la máquina, dado que el objetivo de ésta es ganar la partida y sería una contradicción plantear una jugada desfavorable para ella).

Capítulo 2

Búsqueda

¿Cómo se diseña un nuevo fármaco? Hasta hace muy poco las empresas farmacéuticas aún lo hacían completamente a mano, es decir, sin ninguna otra ayuda tecnológica más que un lápiz y un papel. Diseñaban y optimizaban la estructura química del fármaco y, mediante el uso de laboratorios especializados, sintetizaban y probaban cada versión mejorada para comprobar si su efectividad aumentaba o no. Este proceso completamente manual de prueba y error es lo que, en parte, provoca que el coste de diseñar un nuevo medicamento ronde los 1.000 millones de dólares de media.

Cuando se diseña un nuevo fármaco, por lo general se está diseñando una molécula que pueda interaccionar con una proteína y, eventualmente, inhibir su comportamiento. Las proteínas actúan en los seres vivos desencadenando lo que se conoce como «cascadas metabólicas», que son una sucesión de reacciones bioquímicas mediadas por las mismas proteínas. Por tanto, si una molécula inhibe el comportamiento de una de las proteínas involucradas en la ruta metabólica de interés, esta molécula interrumpirá dicha ruta y podría ser un buen fármaco.



En este escenario simplificado, el fármaco impide la interacción de la proteína diana con la proteína c, interrumpiendo la ruta metabólica.

Para conseguir inhibir una de estas proteínas se debe conseguir que el fármaco se una a ella de una manera determinada. Por tanto, gran parte del esfuerzo de di-

seño cuando se desarrolla un nuevo fármaco es que esta molécula se una al centro activo de la proteína de interés, también conocida como diana terapéutica.

Para saber si una molécula se une eficientemente a una proteína se debe medir la energía de la interacción. La energía de la interacción entre una molécula candidata a convertirse en fármaco y su proteína diana es aquella energía que debe aplicarse al sistema para que éste se mantenga unido. Por ejemplo, si quisiéramos unir un imán de nevera a la puerta de dicho electrodoméstico, no sería necesario aplicar ninguna fuerza de forma continua, ya que el imán tiende a unirse al metal de la puerta debido a sus propiedades magnéticas. Incluso en el caso de que el imán sea realmente potente, sólo con acercarlo a la puerta ya notamos cómo es atraído con cierta fuerza hacia ella. En este caso, estaríamos hablando de que la energía que debe aplicarse para mantener unidos imán y puerta es negativa, ya que, de hecho, ambos objetos se atraen.

Como se podrá suponer, un fármaco que no se sienta atraído por su proteína diana no tiene valor como tal, ya que se limitará a flotar por el riego sanguíneo o los tejidos, ignorando y siendo ignorado por la proteína que debería ser inhibida. Así pues, cuando se busca un buen candidato a fármaco se pretende encontrar un compuesto cuya energía de interacción sea lo más negativa posible, ya que esto es una muestra de hasta qué punto se verán atraídos fármaco y proteína. Por tanto, el principal objetivo que se debe optimizar en un proceso de diseño de un nuevo fármaco es la minimización de esa energía de unión.

Este tipo de problemas cuya solución consiste en la determinación de unos parámetros óptimos (ya sea determinar qué pieza del tablero de ajedrez se debe mover en cada turno con el fin de ganar la partida, o bien las dimensiones y posiciones que debe tener cada una de las vigas que conforman un puente para minimizar su coste y maximizar su resistencia) se agrupan bajo la categoría de «problemas de búsqueda». La búsqueda es una de las grandes áreas de la inteligencia artificial. En un ejercicio de búsqueda, a menudo se deben encontrar los parámetros que maximizan una función matemática y, en este caso particular esa búsqueda también es conocida como «optimización».

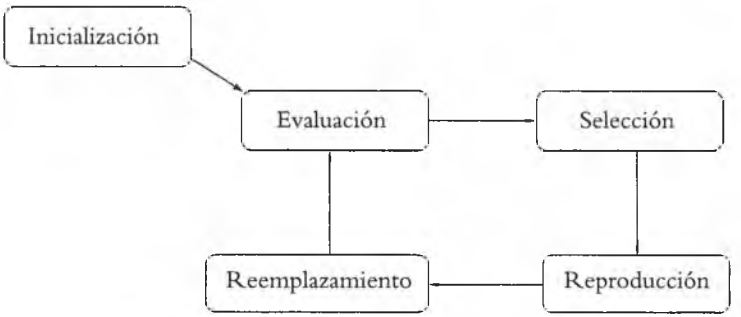
Darwin ya lo dijo

Una de las técnicas más utilizadas para resolver problemas de búsqueda es la computación evolutiva. Del mismo modo que la naturaleza ha sabido hacer evolucionar a sus seres vivos para optimizar su supervivencia en sus respectivos medios naturales,

la computación evolutiva utiliza mecanismos similares para optimizar funciones de diversa complejidad.

La computación evolutiva fue inicialmente propuesta por el investigador John Holland en 1975 en su libro *Adaptation in Natural and Artificial Systems (Adaptación en sistemas naturales y artificiales)*, aunque la ciencia occidental descubrió más tarde que los ingenieros alemanes ya usaban estas estrategias para optimizar las toberas de los primeros motores a reacción de los aviones en la Segunda Guerra Mundial. La computación evolutiva abarca una gran colección de técnicas o algoritmos evolutivos completamente inspirados en las leyes de la evolución natural propuestas por Darwin, según las cuales los individuos mejor preparados son los que mejores índices de supervivencia presentan y, por tanto, los que más descendientes tienen.

En esta metáfora inspirada en las leyes de la evolución natural se cuenta con poblaciones de individuos en las que cada uno de ellos representa una posible solución a un problema. Y, ya sea una buena o una mala solución, lo que tratan de hacer los algoritmos evolutivos después de evaluar la bondad de cada uno de los individuos es seleccionar los mejores para, a partir de ellos, dar lugar a una segunda generación. Siguiendo un proceso iterativo, los individuos de las sucesivas generaciones son evaluados, seleccionados y cruzados de manera que se obtienen poblaciones o generaciones nuevas cada vez. Este proceso se termina según criterios de parada que pueden variar dependiendo del problema. Así pues, un algoritmo evolutivo se basa en cinco grandes etapas: inicialización, evaluación, selección, reproducción y reemplazamiento, tal como se muestra en el siguiente esquema:



Las diferencias entre un algoritmo evolutivo y otro vienen determinadas por las diversas maneras en que se implementa cada una de estas grandes etapas.

DARWIN Y LAMARCK: DOS VISIONES DISTINTAS SOBRE LA EVOLUCIÓN

Jean-Baptiste-Pierre-Antoine de Monet, Caballero de Lamarck (1744-1829), fue un naturalista francés que revolucionó la biología con importantes aportaciones, como la clasificación de los seres vivos según su complejidad o la determinación de una división clara entre el mundo orgánico y el inorgánico. Otra de sus aportaciones a la ciencia fue la elaboración de la primera teoría de la evolución biológica, descrita en su obra *Philosophie Zoologique (Filosofía zoológica)* en 1809, cincuenta años antes de que se diera a conocer la teoría de la evolución de Darwin.

La teoría lamarckiana, al contrario que la darwinista, se basa en la «herencia de los caracteres adquiridos», esto es, en la capacidad de los individuos de trasladar a su descendencia las adaptaciones al medio que ellos hayan realizado en vida. Un buen ejemplo en el que se distinguen perfectamente los dos puntos de vista con respecto a la evolución es el largo cuello de las jirafas. Según Lamarck, la jirafa que estiró más el cuello y logró desarrollar mejor los músculos para llegar a las hojas más altas pasó esta característica a su descendencia, que a su vez siguió desarrollando tales músculos y pasándoselos a su descendencia hasta que el cuello alcanzó el tamaño actual. Según la teoría darwinista, en cambio, fue la jirafa que nació con un cuello más largo o con unos músculos más potentes la que logró pasar dicha característica a su descendencia, independientemente de los esfuerzos que realizara en vida.

Aunque las hipótesis de Lamarck fueron desechadas como falsas en favor de las de Darwin, recientemente se les han reconocido cierta veracidad en algunos casos concretos. Por ejemplo, se sabe que una madre que haya superado una enfermedad para la que ha generado anticuerpos, puede transmitir estos anticuerpos a su descendencia, de modo que sus hijos serán inmunes también a dicha enfermedad. Así pues, estaríamos ante un caso de transmisión de caracteres adquiridos en vida como adaptación al medio.

Caricatura de Lamarck caracterizado como una jirafa.

La inicialización

La inicialización de la población es una etapa bastante independiente del algoritmo evolutivo que se use; de hecho, depende más de las características del problema en cuestión. Hay problemas en los que existen restricciones que deben tenerse en cuenta; otros en los que no se conoce absolutamente nada sobre cómo tiene que ser una buena solución y, por tanto, el problema se inicia de manera completamente aleatoria, y otros en los que se prefiere que la inicialización sea aleatoria, pero postulando que los individuos generados en esta primera generación tengan una cierta diversidad garantizada, para así estar seguros de que no se deja ningún espacio por explorar.

La decisión de cómo debe representarse el conocimiento dentro de un individuo es especialmente importante en esta etapa, ya que determinará en gran medida el resto del algoritmo evolutivo. Una de las representaciones más comunes se realiza mediante cromosomas, un nuevo concepto inspirado en la naturaleza: un cromosoma es una secuencia de genes, y cada gen es un número que representa parte de una solución.

Pongamos como ejemplo un algoritmo que busca maximizar la capacidad de una caja de cartón minimizando la cantidad de cartón usado para fabricarla. Si se utiliza un algoritmo evolutivo, los cromosomas que representarían la solución tendrían tres genes: longitud, amplitud y altura. Por tanto, en la etapa de inicialización se crearía una población de cajas aleatorias representadas por ternas de números dentro de los rangos permitidos, y el algoritmo iría evolucionado las poblaciones de cajas hasta encontrar la caja óptima según los criterios establecidos.

La evaluación

Después de la inicialización viene la etapa de evaluación, que suele decirse que es la más importante del proceso, ya que define el problema que debe resolverse. El primer paso de la evaluación es reconstruir la solución, es decir, para cada individuo se toma la información de su cromosoma (genotipo) para simular la solución representada (fenotipo). Este proceso puede albergar distintos grados de complejidad, desde el simple cálculo del volumen de una caja conociendo sus dimensiones, como ocurría en el problema de la caja de cartón, hasta cálculos extremadamente costosos y complejos, como la simulación de la resistencia de un puente en su proceso de diseño.

Una vez que se tiene el fenotipo reconstruido, se procede a evaluar la bondad de esta solución y se le asigna un valor de bondad a cada individuo, valor que en posteriores etapas evolutivas será usado para discernir entre las buenas y las malas soluciones. De nuevo, el propio proceso de evaluación de fenotipos puede ser complejo, costoso e incluso ruidoso; es decir, en la resolución de algunos problemas complejos, un mismo fenotipo evaluado diversas veces no tiene por qué dar siempre el mismo grado de bondad. El ruido, al cual también podríamos llamar «error», es una constante en problemas en los que la evaluación de la bondad pasa por realizar simulaciones numéricas. Por ejemplo, en la simulación de la resistencia a la fatiga de una pieza de un motor de combustión, resolver las ecuaciones matemáticas que determinan dicha fatiga sería tan costoso que lo mejor es pasar por un proceso simulador, el cual, probablemente, nos llevará a resultados algo diferentes en cada simulación independiente de la misma pieza.

En unos resultados presentados por la empresa Honda en el año 2004 en los que se habían usado algoritmos genéticos para diseñar piezas de motores de combustión, el proceso de evaluación no sólo era ruidoso y un tanto inexacto, sino también lento, ya que se tardaba ocho horas en obtener el resultado de la bondad calculada de cada individuo de la población.

EL PÁJARO RECHONCHO DE LAS ISLAS MAURICIO Y LA PRESIÓN EVOLUTIVA

Cuando los exploradores llegaron por primera vez a las islas Mauricio, en el siglo XVII, se encontraron con un inesperado regalo de los cielos: un pájaro rechoncho, de carne suculenta, con unas alas demasiado pequeñas para permitirle alzar el vuelo y unas patas demasiado cortas para poder huir, al que los hombres llamaron «dodo». Lo cazaron sin piedad y las mascotas de los hombres (perros y gatos), así como otras nuevas especies introducidas en las islas, como las ratas, destruyeron sus nidos para comerse sus huevos. El pobre dodo se extinguió en poco menos de un siglo y actualmente sólo nos quedan dibujos y grabados de cómo debió de ser esta simpática e inofensiva ave. El dodo no había experimentado nunca la «necesidad» de evolucionar, nunca había estado sometido a una presión evolutiva, y cuando ésta apareció ya no tuvo tiempo de hacerle frente. La presión evolutiva es el motor de la evolución. Sin una cierta presión evolutiva los seres vivos no tienen razones suficientes para adaptarse al medio, de modo que no tienen la necesidad de desarrollar una forma, un comportamiento o un aspecto óptimos. A lo largo de la historia de las ciencias naturales se han ido documentando especies que estaban claramente en esta situación, es decir, especies

La selección

La siguiente etapa de un algoritmo evolutivo, una vez que se han evaluado los individuos de la presente generación, es la selección. La idea última de la selección es escoger los mejores individuos que se reproducirán para generar la descendencia que formará la siguiente generación. Este proceso de selección de los mejores individuos es también la base de la evolución natural y se denomina «presión evolutiva». La presión evolutiva es mayor en función de cuán reducido es el porcentaje de individuos que pasan a la siguiente generación. Sin embargo, si se aplica una estrategia tan sencilla como coger directamente los mejores individuos, se puede demostrar que la presión evolutiva aplicada es demasiado elevada y, ante presiones evolutivas excesivas, los algoritmos evolutivos no suelen funcionar bien, ya que suelen caer en máximos locales.

La principal utilidad de un algoritmo evolutivo es poder encontrar soluciones buenas en grandes espacios de búsqueda, o, dicho en términos matemáticos, encontrar los óptimos de funciones, normalmente con muchas dimensiones y multimodales, es decir, funciones con diversos máximos locales o globales. Si la presión aplicada a la optimización evolutiva es excesiva, esto es, si se trata de encontrar la

inmersas en un entorno con abundancia de alimento, ausencia de depredadores o poca competencia entre especies, hecho que ha dificultado la aparición de ciertas propiedades indispensables para otras especies similares en entornos más competitivos.

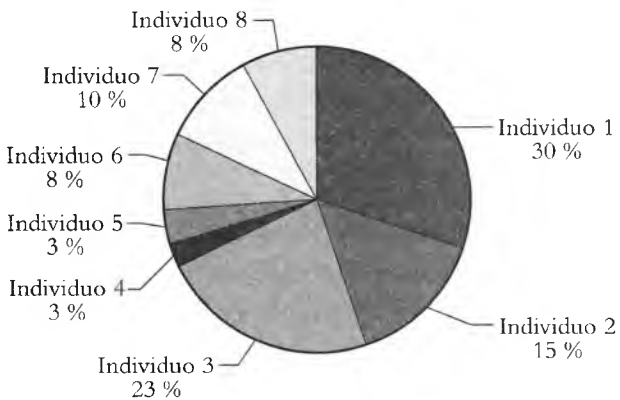
Se podría decir que eso es lo que le ocurrió al dodo. Sin depredadores ni escasez de alimentos en su protegido ecosistema isleño, no tuvo necesidad de desarrollar unas alas útiles o unas patas capaces de dotarlo de velocidad. De hecho, la traducción literal del portugués de la palabra *dodo* es «estúpido». ¿Tal vez fue la ausencia de presión evolutiva la que convirtió a este animal en un «estúpido»?

Dodo en un grabado del siglo XVII.

solución demasiado deprisa seleccionando enseguida los mejores individuos sin explorar más allá, el algoritmo estará convergiendo prematuramente y estará cayendo en máximos locales.

La selección es la etapa ideal en la que se puede modular la presión evolutiva de un algoritmo evolutivo. El extremo donde esa presión sería más elevada se tendría en el caso de que se realizara una selección unitaria, es decir, seleccionando sólo el mejor individuo de la población para dar lugar con él a los individuos de la generación siguiente. En el otro extremo estaría la selección aleatoria, aquella en la que no se tiene en cuenta la bondad de los individuos. Lógicamente, la estrategia que conviene seguir será un punto medio entre las dos, donde se intenten seleccionar los mejores individuos para que se reproduzcan, pero introduciendo siempre cierto grado de diversidad para explorar otros caminos. Siguiendo esta estrategia, un individuo, por malo que sea, siempre tendrá alguna probabilidad de ser seleccionado aunque en la población se encuentren presentes individuos mucho mejores. Las tres estrategias de selección que se comportan de esta manera, y que son, además, las más utilizadas en la actualidad, son: la ruleta, la selección basada en el rango y el torneo.

La estrategia de selección de la ruleta es un sistema bastante sencillo en el que cada individuo tiene una probabilidad de ser seleccionado proporcional a su bondad en relación a las bondades del resto de los individuos. Por tanto, si se tienen que seleccionar diez individuos, se hace girar la ruleta diez veces.



En el ejemplo de la figura hay ocho individuos y cada uno tiene una bondad proporcional al total indicado en cada porción. Como se puede intuir, cada vez que se hace girar la ruleta, la probabilidad de que un individuo determinado sea selec-

cionado es proporcional a esta bondad con respecto al total pero, además, la estrategia de la ruleta no excluye la selección de individuos con menor bondad, sino que simplemente hay menos probabilidad de que esto ocurra. Si se hace rodar la ruleta diez veces, seguro que se selecciona en diversas ocasiones a individuos buenos, pero también es probable que alguna de las veces se seleccione un individuo poco competitivo. Esta posibilidad de seleccionar individuos poco competitivos es lo que dota a los algoritmos genéticos de tanta potencia, ya que les permite seguir distintas pistas al mismo tiempo, de modo que pueden descubrir otros espacios, por lo que son capaces de identificar un gran número de distintos máximos y encontrar, a largo plazo, un buen máximo local o, en el mejor de los casos, el máximo global.

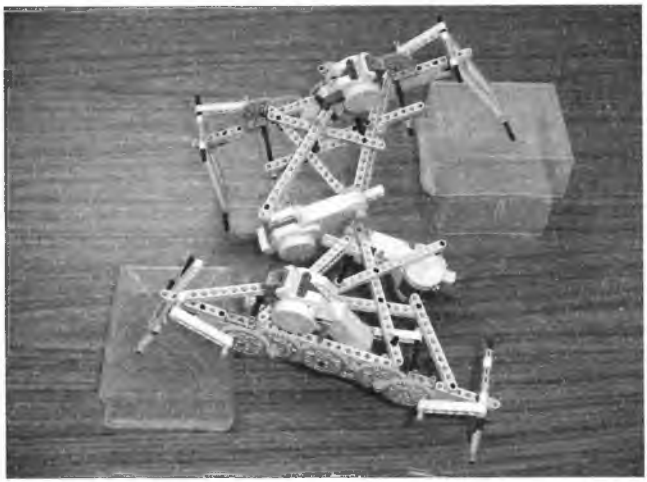
Otro sistema de selección adecuado para la resolución de problemas complejos es la selección basada en el rango. El sistema vuelve a ser bastante sencillo: se trata de seleccionar n copias del mejor individuo, $n-1$, del segundo individuo, y así hasta que se llega a $n=0$. Con este sistema se elimina la posibilidad de que un superindividuo eclipse las posibilidades de selección de cualquier otro individuo. Se conoce como «superindividuo» a aquel individuo que, aun estando lejos de un óptimo, es mucho mejor que sus compañeros de generación y, por tanto, la población se estanca alrededor de él y el algoritmo no es capaz de mejorar sus cualidades.

Sin embargo, es el tercer método, el del torneo, el que finalmente ha conseguido el monopolio entre las políticas de selección usadas en la resolución de problemas reales por sus buenas propiedades matemáticas y por la alta versatilidad que ofrece en la modulación de la presión evolutiva. El torneo funciona del mismo modo que los emparejamientos en una competición deportiva. Se seleccionan emparejamientos aleatorios de individuos, de dos en dos, y se considera que el mejor es aquel que gana el torneo y queda seleccionado. Por tanto, en este caso deben realizarse tantos emparejamientos como individuos se necesite seleccionar. Pero ¿por qué se dice que el método del torneo es tan versátil en la modulación de la presión evolutiva? Bien, ¿qué pasaría si en vez de organizar torneos de dos individuos se hicieran de n individuos? ¿Y si en vez de haber un solo ganador por torneo hubiera m? En este caso se dice que se están organizando torneos de n:m y, cuanto mayor sea n, mayor presión evolutiva se ejerce, y cuanto mayor sea m, menor es la presión evolutiva.

Para entender mejor el esquema del torneo, piénsese en las liguillas iniciales de la UEFA Champions League. En ese caso, los torneos son 4:2, es decir, se seleccionan aleatoriamente cuatro equipos de fútbol y sólo los dos mejores son seleccionados para pasar a la siguiente fase de la competición. En realidad, en el caso de la Champions League no se puede hablar de un torneo estrictamente aleatorio, ya que

rigen algunas normas en la selección de estos cuatro equipos iniciales, como que no coincidan dos del mismo país en las primeras fases. De la misma manera, en los algoritmos evolutivos podemos también poner nuestras normas, lo que derivará en un tipo de evolución o en otra.

Una norma bastante usada es la de que sólo compitan en un mismo torneo los individuos más parecidos entre sí. De esta manera, el algoritmo es capaz de optimizar funciones con muchos óptimos.



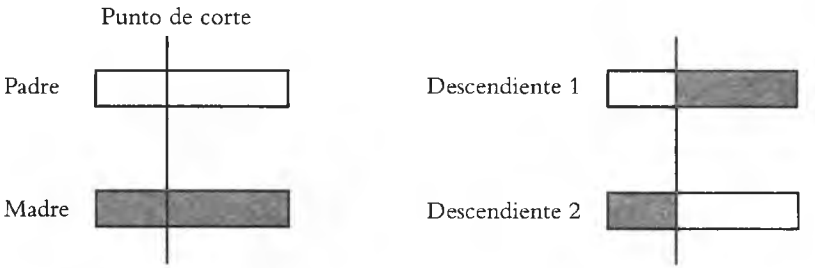
Estos «cangrejos-robot» son buscadores de luz. Uno carece de patas, mientras que el otro tiene cuatro. Ambos fueron utilizados por Josh Bongard, de la Universidad de Vermont, quien los dotó de un algoritmo genético evolutivo y pudo comprobar que sus máquinas, una vez evolucionadas, funcionaban mejor que los robots normales ideados expresamente para el mismo fin.

La reproducción

Una que vez se han seleccionado los individuos que van a tener descendencia, llega la etapa de la reproducción. Existen varios tipos de sistemas de reproducción y, aunque ésta no es necesariamente la parte más importante de un algoritmo evolutivo, en realidad éste se define por su sistema de reproducción; es decir, un algoritmo evolutivo concreto recibe su nombre en función del tipo de reproducción que utiliza. Por ejemplo, los algoritmos genéticos, de los que hablaremos a conti-

nuación, son los algoritmos evolutivos que tienen un sistema de reproducción de cruce con mutación.

Los algoritmos genéticos son los algoritmos evolutivos más usados gracias al buen compromiso que consiguen entre dificultad de programación y buenos resultados obtenidos con ellos. La reproducción por cruce y mutación está basada fuertemente en los conceptos de la genética. En un algoritmo genético cada uno de los individuos está representado mediante un cromosoma, y cada cromosoma es una secuencia de genes. Cuando se cruzan los cromosomas de los progenitores, primero se crea un punto de corte aleatorio que los divide en dos mitades. A continuación, estas cuatro mitades (dos por cada progenitor) se cruzan entre sí para generar dos descendientes: el primer descendiente contiene la primera parte del cromosoma del primer progenitor (llamémosle padre) más la segunda parte del cromosoma del segundo progenitor (madre), y el segundo descendiente consta de un cromosoma formado por la primera parte del cromosoma de la madre, hasta el punto de corte, y por la segunda parte del cromosoma del padre.



Para concluir con los algoritmos genéticos, una vez que los descendientes han sido creados, se pasa por un proceso de mutación en el que, mediante una probabilidad muy baja (típicamente alrededor del 5%), se cambian de modo aleatorio valores de los genes que componen los nuevos cromosomas. Tanto en la práctica como en la teoría se puede demostrar que, sin la mutación, los algoritmos genéticos no son buenos motores de optimización, ya que se suelen estancar en subóptimos o máximos locales. La mutación permite a los algoritmos genéticos ir dando pequeños saltos aleatorios dentro del espacio de búsqueda. Si los resultados de estos saltos aleatorios no son prometedores, se perderán en el proceso evolutivo, pero si son positivos, serán adoptados por los buenos individuos de las generaciones venideras.

GREGOR MENDEL Y LA GENÉTICA

Gregor Mendel (1822-1884) fue un monje austriaco que descubrió y publicó, en 1866, las primeras leyes existentes sobre genómica. Estas leyes, conocidas en la actualidad como leyes de Mendel, describían la transmisión de ciertas características de padres a hijos basándose en un estudio que cruzaba distintas especies de guisantes. Estas leyes introdujeron un concepto esencial para la genética y para la ciencia en general: el de la existencia de genes dominantes y genes recesivos.

Mendel inició sus experimentos realizando observaciones sobre la coloración de las semillas que producían distintas plantas de guisantes. La primera generación la obtuvo a base de cruzar una planta que producía semillas de color amarillo con otra que las generaba de color verde. Entonces observó que las plantas resultantes de dicho cruce producían sólo semillas amarillas. Sin embargo, más tarde observó que, aunque al cruzar estas plantas entre sí la generación resultante producía en su mayoría semillas amarillas, sorprendentemente, algunas plantas volvían a generar semillas verdes. La relación entre plantas de semillas amarillas y verdes era de 3:1. Después de realizar experimentos similares con otras características, finalmente Mendel llegó a la conclusión de que existían unos genes que eran dominantes sobre otros, de modo que oscurecían la existencia del gen dominado y no le permitían expresarse en el individuo. Esto explicaba que del cruce de individuos con el mismo gen expresado pudieran darse descendientes en que el gen expresado fuera otro, ya que, sin saberlo, ambos padres poseían dicho gen, aunque «oscurecido» por el dominante.

A pesar de que en su momento no tuvieron gran trascendencia, los trabajos de Mendel describen la base de la genética, el área de la ciencia que estudia los genes y la transmisión de las características de padres a hijos, y que ha sido absolutamente determinante para la medicina moderna.

El reemplazamiento

Por último, la etapa que cierra el ciclo del proceso evolutivo es la del reemplazamiento. El objetivo de esta etapa es seleccionar qué individuos de la generación anterior serán reemplazados por los nuevos individuos generados en la reproducción. La estrategia más habitual es reemplazar todos los individuos de la anterior generación excepto el mejor, al que se le da la oportunidad de «vivir» durante otra generación más. Este método, conocido como «elitismo», ha demostrado que, aun siendo un proceso muy simple y no muy natural, tiene una potencia extraordinaria.

Sin embargo, se han propuesto muchas otras estrategias para reemplazar individuos. Nótese que, de nuevo, al igual que en la etapa de selección, según como se seleccionen los individuos que van a ser reemplazados, se puede modular la presión del proceso evolutivo: si siempre se seleccionan todos los individuos de la población y se reemplazan por los nuevos individuos, no se aplica ninguna presión evolutiva, y al contrario, si sólo se seleccionan los malos individuos de la población anterior para ser reemplazados, la presión aumenta enormemente.

ALGORITMOS EVOLUTIVOS LAMARCKIANOS

La dualidad evolución darwiniana-evolución lamarckiana también existe en los algoritmos evolutivos y, de hecho, ambos métodos han demostrado ser altamente eficaces para resolver problemas de optimización numérica. Los algoritmos evolutivos darwinianos son los habituales, los que se han ido describiendo a lo largo de este capítulo, mientras que los lamarckianos incluyen un paso adicional entre la evaluación y la selección. Este paso consiste en una corta optimización local que simula el aprendizaje o la adaptación de un individuo al medio antes de tener descendencia.

Esta etapa de optimización local normalmente se basa en pequeñas mutaciones que se aplican a cada individuo. Luego se vuelve a evaluar la bondad del individuo para ver si la mutación ha introducido una mejora. Si es así, ésta se acepta y se repite de nuevo el ciclo de mutación-evaluación, pero si la mutación ha empeorado la bondad del individuo, ésta se descarta y se vuelve a repetir el ciclo de mutación-evaluación desde el estado anterior a la mutación actual. Los primeros algoritmos evolutivos lamarckianos fueron bautizados como «estrategias evolutivas» y, como ya se ha dicho, los alemanes los usaron durante la Segunda Guerra Mundial como herramienta para optimizar las toberas de los motores de reacción de sus aviones.

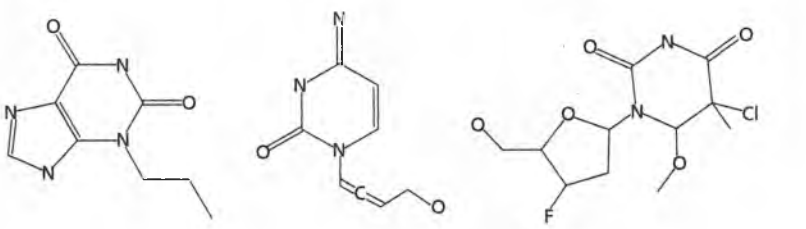
Por otro lado, en esta etapa también se pueden aplicar de modo eficiente políticas de especiación, es decir, métodos que faciliten la identificación de varias soluciones para aquellos problemas que tengan diversos óptimos. El método más usado en este caso es el reemplazamiento mediante *niching*. Esta estrategia consiste en seleccionar, para cada nuevo individuo generado, los individuos de la población anterior que más se asemejen a él. En la siguiente generación, sólo el mejor individuo del grupo de similares podrá permanecer en ella.

Hasta el momento se han explicado algunos de los métodos más habituales para llevar a cabo cada una de las etapas evolutivas. Sin embargo, el lector debe saber que existen un sinfín de métodos para realizar cada una de estas fases evolutivas.

Un ejemplo práctico: evolucionando hacia un buen fármaco

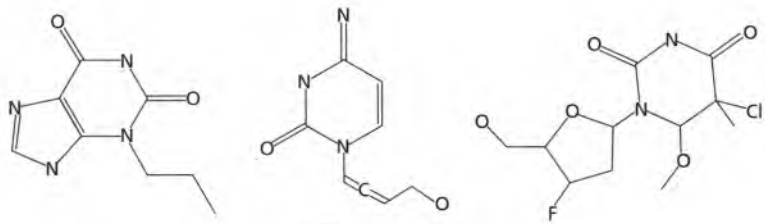
Como ya se ha visto, la inteligencia artificial ofrece métodos de optimización basados en procesos naturales que obtienen grandes resultados. Y, recientemente, la computación evolutiva ha penetrado en el nicho científico del diseño de fármacos con un éxito apreciable. Recordemos que en el diseño de medicamentos, el objetivo es lograr un compuesto cuya energía de unión con una proteína determinada sea lo más negativa posible, de modo que una atracción irresistible las junte en el interior de nuestro organismo como un caramelo y un niño goloso.

Veamos cómo actuaría un algoritmo evolutivo en el proceso de optimización de un candidato a fármaco. En primer lugar, el algoritmo tiene que inicializar la población de moléculas. En esta etapa lo que suele hacerse es proponer una generación aleatoria de moléculas. Para simplificar el ejemplo, haremos generaciones de sólo tres moléculas, aunque normalmente las generaciones contienen centenares de ellas:



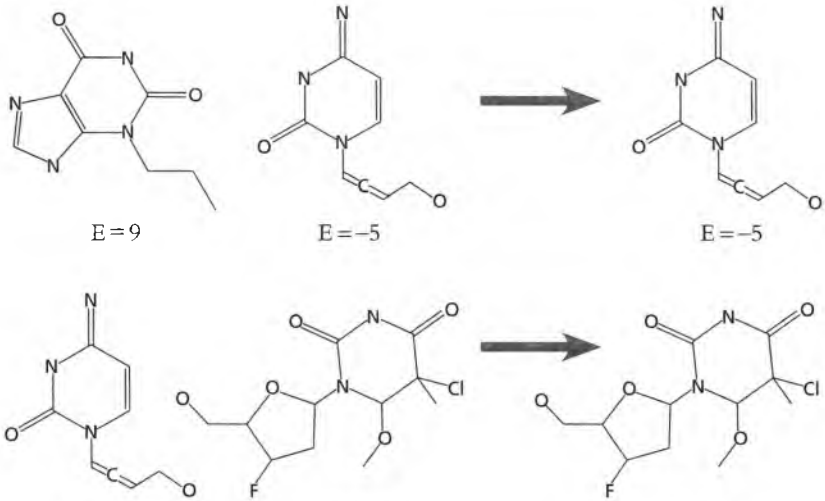
A continuación, se deben evaluar estas moléculas iniciales mediante la estimación de la energía de interacción de cada una de ellas con la proteína diana. Para ello se pueden utilizar diversas herramientas computacionales. Una de ellas (que

sólo mencionaremos, pero no explicaremos) es el *docking*, un proceso de simulación tridimensional que prevé cuál será el comportamiento de la molécula cuando se encuentre con su diana: si va a poder encajar en ella o no y cuál será la energía de la unión. De este modo, se da la curiosa circunstancia de que mientras usamos un algoritmo evolutivo para encontrar nuestra molécula perfecta, también estamos utilizando un algoritmo evolutivo para evaluar cuán buena es esta molécula con respecto a las demás. Tras el *docking* ya tenemos las moléculas evaluadas:



E = 9 E = –5 E = –8

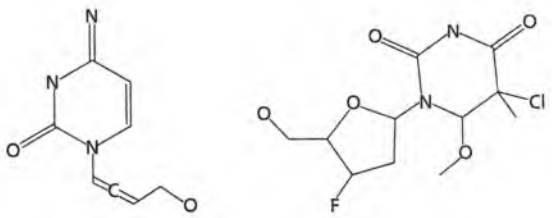
La siguiente etapa es la selección, que organizaremos, por ejemplo, a través de un torneo molecular, en el que se seleccionarán parejas de moléculas al azar, se compararán sus energías de interacción y se decidirá si éstas «se quedan» o «se van». Recordemos que las energías de interacción deben ser lo más negativas posible.



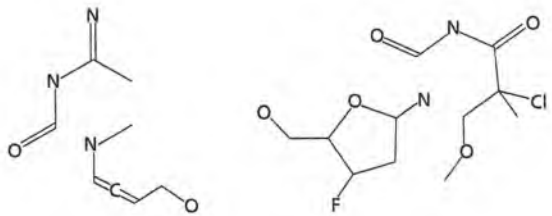
E = 9 E = –5 E = –5

E = –5 E = –8 E = –8

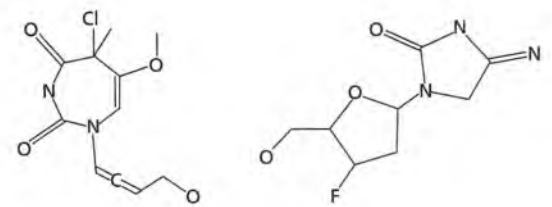
El siguiente paso evolutivo es la reproducción, donde, a partir de las moléculas seleccionadas, se crean nuevas moléculas que combinan las propiedades de las primeras. Así pues, se cruzan las dos moléculas seleccionadas en el paso anterior para generar dos nuevas moléculas, que serán un cruce de sus progenitores:



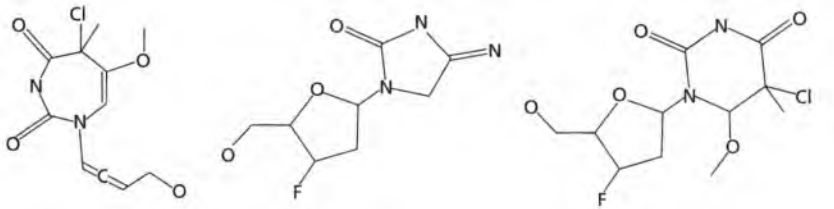
En la siguiente ilustración vemos cómo las dos moléculas se parten en dos secciones:



Y, por último, se originan dos nuevas moléculas uniendo las partes de las anteriores:



Finalmente, en la etapa de reemplazamiento, se retiran los individuos de la primera generación y se reemplazan por los nuevos individuos creados. El método de reemplazamiento más usado, y también el más simple, es el del elitismo, en el que todas las moléculas de una generación quedan reemplazadas, excepto la mejor. En este caso, la nueva generación contiene las dos moléculas generadas a partir del cruce, más la mejor molécula de la generación anterior, que era la que tenía una energía de interacción de –8.



Después del reemplazamiento, el ciclo evolutivo se cierra y se itera tantas veces como sea necesario. Es decir, ahora estas moléculas de segunda generación se evaluarían, después se seleccionarían, etc., y se llegaría a una tercera generación. Y así hasta que se repita un número de generaciones predefinidas o la población haya convergido, es decir, que el 90% de los individuos sean la misma molécula.

Naturalmente, la realidad es bastante más complicada; lo que hemos explicado aquí es una burda simplificación, pero ¿a qué es bonito?

Capítulo 3

Aprendizaje artificial

Jueves 6 de mayo de 2010, 9:30 h de la mañana. Apertura de los mercados de valores estadounidenses. En principio, un día normal de transacciones financieras: la mañana va pasando sin anormalidades aparentes. Pero por la tarde, a las 14:45 h, sin ninguna causa clara, algunos de los valores más importantes del mercado empiezan a desplomarse en cuestión de segundos. Aun con la volatilidad característica de los mercados en aquel periodo de inestabilidad financiera, esta caída fue bastante sorprendente, ya que algunas de las compañías más grandes y sólidas estaban sufriendo bajadas de más del 60%, y todo el mercado de valores norteamericano y, por consiguiente, el mundial, se estaba desmoronando en cuestión de minutos. Aquel día, el Índice Industrial Dow Jones (uno de los índices de referencia bursátil más usados a escala internacional) llegó a caer un 9,2%, el descenso en un solo día más importante de toda la historia, aunque después llegó a estabilizarse en un descenso de «sólo» el 3,2%. En pocos segundos desaparecieron del mercado un trillón de dólares de valor, en lo que hoy en día se conoce como el «Flash Crash».

En el parqué de la Bolsa de Nueva York, en Wall Street, fue donde se detectaron algunos de los primeros indicios del Flash Crash.

Aunque se ha tratado de dar muchas razones para explicar el Flash Crash, aún no se ha determinado una causa clara. Sin embargo, una de las hipótesis que más fuerza ha tomado entre los investigadores financieros fue el impacto de los HFT (High Frequency Traders) o negociadores de alta frecuencia, aunque esta explicación siempre ha sido negada por los reguladores del mercado. Los HFT son sistemas automáticos e inteligentes de compraventa de acciones y productos financieros que son capaces de tomar decisiones y de actuar en cuestión de microsegundos. Se calcula que hoy en día el 50% de las operaciones financieras internacionales están realizadas por sistemas HFT.

Pero ¿cómo puede un sistema informático, inteligente o no, tomar decisiones de tal magnitud de manera tan rápida? Cualquier inversor *amateur* sabrá que los precios de los mercados financieros dependen de un sinfín de variables socio-económico-políticas estructurales y coyunturales, desde las últimas declaraciones del ministro de trabajo finlandés sobre las regulaciones laborales de su país hasta un descenso imprevisto de la demanda de crudo a causa de una subida de las temperaturas en el sur de Alemania. ¿Cómo podría entonces un sistema informático tener en cuenta tanta información como para tomar decisiones aparentemente inteligentes de compraventa de acciones, y además hacerlo en segundos? Ésa es la cuestión. El aprendizaje artificial es uno de los grandes pilares de la inteligencia artificial. Posiblemente no seamos conscientes de ello, pero gran parte de las actividades y escenarios en los que nos vemos envueltos a diario están controlados por completo por máquinas inteligentes. Sin embargo, antes de empezar a operar, éstas deben aprender cómo hacerlo.

Un ejemplo de aprendizaje: la predicción de tumores

La predicción de tumores es uno de aquellos casos en los que la inteligencia artificial puede ser de gran utilidad para los expertos médicos a la hora de tomar decisiones en diferentes etapas de diagnóstico. Someterse a una mamografía es, o debería ser, una práctica regular entre las mujeres adultas para la predicción precoz de cánceres de mama. Una mamografía no es más que una radiografía de la glándula mamaria que permite ver ciertas anomalías en el tejido, algunas de las cuales podrían ser tumores incipientes de mama. Por eso, cada vez que un radiólogo identifica una de estas anomalías en una mamografía, ordena hacer un análisis más exhaustivo que requiere una biopsia, o extracción del tejido, una práctica bastante más agresiva, molesta y cara que la mamografía.

Sin embargo, una vez que se analizan los resultados de la biopsia, sucede que en un 10% de los casos se trataba de un falso positivo, es decir, se apreciaba una anomalía en la mamografía, pero al hacer la biopsia no se identificó ningún rastro de tumor. Por ello sería de gran valor contar con una herramienta que permitiera reducir al máximo ese 10% de falsos positivos, no sólo para ahorrar costes a la sanidad pública, sino también para evitar a las pacientes las molestias causadas por las agresivas biopsias y para reducir su estrés y ansiedad al encontrarse en esa situación.

Por otro lado, también existen los falsos negativos, aquellos casos en que la mamografía no refleja ninguna anomalía, pero, desgraciadamente, sí que existe un tumor. Los problemas sanitarios derivados de la existencia de falsos negativos son evidentes, y de ahí la importancia de que las nuevas herramientas diagnósticas sean eficientes tanto en la reducción de falsos positivos como de falsos negativos. Como se verá a continuación, es bastante más complejo lograr que una herramienta sea capaz de reducir los falsos negativos que los falsos positivos. Y los falsos negativos son mucho más graves.

Imaginemos que un oncólogo tiene que analizar una mamografía de una paciente para determinar si hay indicios de tumor. De manera general, el razonamiento o metodología que sigue puede descomponerse en los siguientes pasos:

1. Observa la mamografía y detecta las características más relevantes con la finalidad de determinar el nuevo problema. El conjunto de las características detectadas permiten describir la situación.
2. Busca mamografías diagnosticadas previamente, tanto por él como en los libros de medicina, que tengan características similares.
3. Emite un diagnóstico teniendo en cuenta los diagnósticos del conjunto de mamografías que había recopilado como similares.
4. Finalmente, si lo considera necesario, pide una segunda opinión a compañeros suyos con la finalidad de validar su diagnóstico.
5. A partir del diagnóstico realizado, toma notas pertinentes para recordar el nuevo caso resuelto, ya que éste le puede ser útil en el futuro.

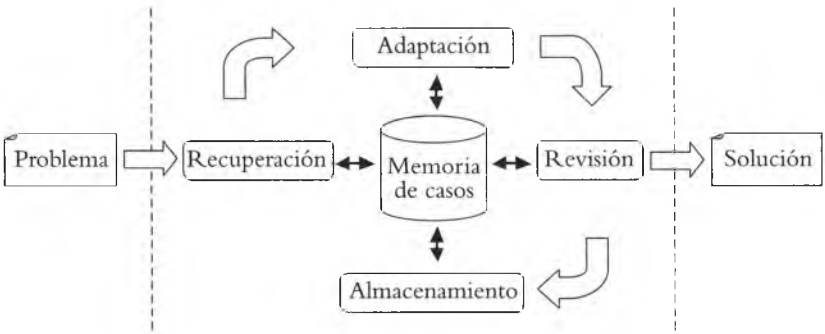
Este procedimiento coincide paso a paso con una de las técnicas de predicción más utilizadas en inteligencia artificial llamada «razonamiento basado en casos», en adelante CBR (Case-Based Reasoning) conforme a sus siglas en inglés. Consiste en resolver nuevos problemas buscando analogías con problemas resueltos de manera satisfactoria en el pasado y, una vez seleccionada la solución más similar, adaptarla a

las características del nuevo problema. Por ello, el CBR no es sólo una herramienta de análisis de datos, sino que con los datos analizados se pretende conseguir un objetivo más general, que es la resolución inteligente de problemas.

CARACTERÍSTICAS USADAS EN LA DETECCIÓN DE TUMORES DE MAMA

El CBR, así como otras técnicas inteligentes, se puede usar para ayudar a diagnosticar la presencia de tumores malignos a partir de mamografías. Como los datos de entrada de cualquiera de estas técnicas son numéricos, se requiere un paso intermedio que haga una extracción automática de esos valores a partir de las imágenes médicas. Concretamente, en el caso de los tumores de mama se suelen extraer diversas mediciones sobre unos elementos frecuentes en las mamas denominados microcalcificaciones, que son unas milimétricas aglomeraciones de calcio en el tejido mamario. Algunas de las características usadas normalmente para la detección de microcalcificaciones malignas en las mamas son: el área de las mismas, su perímetro, la compactibilidad (relación entre el área y su perímetro), el número de agujeros que presentan, la rugosidad (relación entre el perímetro y su irregularidad), la longitud, la amplitud, la elongación (una relación entre la amplitud y la longitud) y la posición del centro de gravedad de la microcalcificación.

De la misma manera que un experto guardaría su experiencia en su memoria o en sus notas, el CBR dispone de una estructura de datos llamada «memoria de casos» donde se almacenan los casos previamente resueltos. El siguiente dibujo resume el funcionamiento del CBR:



La primera fase del CBR, la de recuperación, trata de buscar las soluciones más parecidas al nuevo problema en la memoria de casos del sistema. Aplicado a nuestro ámbito, el objetivo de la fase de recuperación es buscar mamografías diagnosticadas que tengan unas características similares a las de la nueva mamografía que se quiere diagnosticar.

A continuación viene la fase de adaptación, en la que se trata de adaptar la solución similar encontrada a las características del nuevo caso. Por ejemplo, supongamos que una empresa de logística necesita llevar un camión desde Lisboa a Roma y quiere usar CBR para la optimización de la ruta. Lo primero que hará será buscar en su memoria de casos el viaje más parecido realizado hasta el momento. Imaginemos que encuentra otro viaje con una ruta ya optimizada entre Madrid y Milán. Por tanto, gran parte del recorrido se puede reaprovechar y sólo hace falta optimizar la ruta entre Lisboa y Madrid, y entre Milán y Roma. Estas dos optimizaciones de una pequeña parte de la ruta se pueden llevar a cabo mediante otras técnicas informáticas clásicas, pero el caso es que este proceso de adaptación de la ruta Madrid-Milán a la nueva ruta que se planteaba, Lisboa-Roma, es precisamente la fase de adaptación.

Se pasa ahora a la fase de revisión, en la que el experto debe revisar el diagnóstico realizado por la máquina. Es en esta etapa donde hombre y máquina colaboran, lo cual sirve para mejorar día a día el rendimiento de la máquina y, sobre todo, para aumentar la fiabilidad en sistemas en los que la predicción es crítica. En el caso particular de la predicción de tumores, debido a la misma relevancia del tema, es muy difícil que un sistema sanitario acepte dejar la tarea de diagnóstico únicamente a una herramienta automática de estas características sin la participación de un experto médico. Sin embargo, quién sabe si esto podría cambiar en el futuro…

Finalmente, la última etapa del CBR es la fase en la que, una vez resuelto y revisado por el experto, se decide si el caso entrará a formar parte de la memoria de casos, es decir, se decide si es lo suficientemente representativo como para incluirlo dentro del conjunto de mamografías que servirán para diagnosticar tumores en el futuro.

El éxito del CBR (pero también del razonamiento de un experto) se basa en la capacidad de desarrollar correctamente cada una de las cuatro fases del método. Por ello, los aspectos básicos que deben tenerse en cuenta en cada una de ellas son:

— Criterio de recuperación: no todas las experiencias son útiles. Hace falta determinar qué casos de la experiencia pasada se seleccionan ante un nuevo caso para ser usados en la resolución de éste. Para ello, cabe definir métricas o distancias matemáticas que estimen la distancia entre el nuevo caso y los

que se encuentran almacenados en la memoria de casos. Por ejemplo, en el asunto de las mamografías, ante un nuevo caso que resolver, mediante estas métricas matemáticas se determina cuál es la mamografía anterior, ya diagnosticada, que más se parece a la que hay que diagnosticar ahora.

— Criterio de fiabilidad: cada dominio tiene una complejidad implícita y un nivel de riesgo propio, según el «precio a pagar» por equivocarse. En el caso de la detección de tumores es obvio que el precio que se debe pagar por emitir un falso negativo es mucho mayor que el de predecir como peligrosa una anomalía que no lo era. Por ello, es vital establecer mecanismos para la definición de criterios que ayuden a garantizar la fiabilidad de la propuesta.
— Criterio de validación: la validación de una propuesta requiere la intervención de un experto. En el caso de las mamografías, por su criticidad, es normal que esta fase de validación sea ejercida por el experto radiólogo.
— Criterio de mantenimiento del conocimiento: la capacidad de resolver problemas está estrechamente ligada a la experiencia de la que se dispone. Por eso hace falta asegurarse bien de la consistencia de ese conocimiento, tanto incluyendo casos nuevos que se hayan resuelto como eliminando aquellos que confunden al sistema.

Todos los puntos anteriores tienen el mismo denominador común: la experiencia del sistema almacenada en la memoria de casos. Lo deseable sería que esta memoria de casos se caracterizara siempre por las propiedades de ser:

— Compacta: no debe contener casos redundantes ni con ruido porque éstos pueden distorsionar la realidad y confundir al sistema en el proceso de recuperación de los casos más similares.
— Representativa: no es posible resolver aquello de lo que no se tiene constancia; así pues, es necesario disponer de casos representativos de todos los diferentes aspectos característicos del dominio para no tener una visión parcial de la realidad.
— Reducida: la velocidad con la que responde el sistema está relacionada con el número de elementos de los que se dispone. El tamaño de la memoria tiene que permitir la respuesta del sistema en un tiempo razonable.

Estas tres propiedades pueden resumirse en la premisa siguiente: disponer de un conjunto mínimo de casos independientes capaces de representar completamente el dominio.

Otro ejemplo: el marketing online

Desde la popularización de Internet, la disciplina del marketing ha cambiado por completo. De hecho, hoy en día un elevadísimo porcentaje del marketing diario ha dejado de ser estático y se ha pasado de un marketing de masas a otro personalizado. Por ejemplo, cuando se visita una página web o, en general, se navega por Internet, van apareciendo anuncios, o *banners,* arriba, abajo o en los laterales de las páginas visitadas. Estos anuncios no son en absoluto aleatorios ni estáticos, sino que mediante diversas herramientas se va siguiendo y analizando el patrón de comportamiento del navegante y los anuncios que éste ve están completamente personalizados en función de sus intereses actuales.

¿Quién no ha recibido un correo electrónico a través de Gmail, el servicio de correo de Google, y no ha observado que en los laterales siempre aparece publicidad relacionada con el contenido del mensaje que en ese momento se está leyendo? ¿O quién no ha visitado una página web para consultar alguna cosa y se ha encontrado con anuncios de hoteles en París, cuando justamente es lo que estuvo buscando la semana anterior?

Todos los mecanismos usados por Google, y por otras empresas similares, para dirigir el marketing online son herramientas inteligentes que, de manera instantánea y automática, pueden tomar esas decisiones de publicidad sin la intervención humana. De hecho, si se incorporara algún tipo de intervención humana sería imposible realizar tantas acciones de marketing por segundo, ya que el número de páginas web visitadas por segundo en todo el mundo es una cifra que debe rondar las decenas de millones.

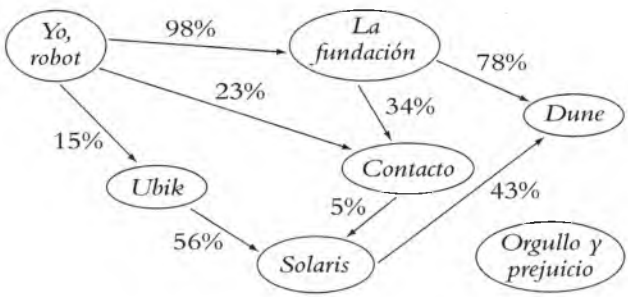
Si se tuviera que escoger la herramienta de marketing online más inteligente de entre todas las posibles, muchos optarían por el mecanismo de sugerencia de libros que usa Amazon, que, dicho sea de paso, es el mismo que utilizan otras empresas para propósitos similares, como, por ejemplo, Yahoo en su Radio LAUNCHcast, que se basa en las canciones que el usuario ha puntuado positivamente para crear su perfil y pincha en el futuro canciones que otros usuarios de un perfil similar al suyo también han escuchado y puntuado positivamente. En Amazon este sistema se puede apreciar claramente cada vez que se busca un objeto determinado, tanto si se es un usuario registrado como si no, en la sección *Customers who bought this item also bought...* («Clientes que compraron este objeto también compraron...»). Aunque pueda parecer trivial, la complejidad que se esconde tras esta simple idea es muy importante, hasta el punto de que esta herramienta se basa en técnicas clasificadas

dentro de la inteligencia artificial, que van bastante mas allá del mero hecho de explorar qué hay en las cestas de la compra de otros usuarios que compraron el mismo objeto que se está revisando ahora.

Página de búsqueda de afinidades de Amazon.

La herramienta típica utilizada para aproximar este tipo de problemas es lo que se conoce como «redes bayesianas» y, de hecho, el mayor centro de investigación mundial especializado en esta herramienta es el Microsoft Research Institute, donde se estudia la aplicabilidad de esta técnica no sólo para el marketing online sino también para otros aspectos, como que la interfaz de usuario de Windows se adapte automáticamente a cada individuo según su manera de trabajar o sus preferencias.

La idea que subyace detrás de una red bayesiana es que hay cadenas de eventos que suelen sucederse y que pueden compartir probabilidades con otras cadenas de eventos; por eso se llaman «redes», porque son cadenas de probabilidades entrecruzadas. Veamos un ejemplo sobre la compra de libros:



OTROS USOS DEL MARKETING AUTOMÁTICO

El marketing en supermercados tiene como objetivo acertar nuestros gustos y necesidades para que, de este modo, llenemos más fácilmente nuestro carro de la compra. El marketing automático no sólo se aplica en el mundo virtual de Internet: hoy en día los bancos, los operadores de telecomunicaciones e incluso los supermercados de barrio son usuarios de esta nueva aproximación. Por ejemplo, ¿quién no conoce los cupones de descuento del supermercado en el que muchas veces hacemos la compra semanal? Lógicamente, no nos suelen dar cupones de descuento para los productos que ya compramos habitualmente (suponiendo, claro está, que lo hagan bien), sino para productos que nosotros no solemos adquirir pero que otros clientes con cestas de la compra similares a la nuestra sí que compran. Con ello, nos dan a conocer ese producto, que posiblemente no conocíamos o no habíamos comprado nunca, para que a partir de esta primera compra tal vez se convierta en un habitual de nuestra cesta. Lo mismo sucede con otro tipo de empresas, como los servicios financieros o de telecomunicaciones, de los que es bastante habitual recibir ofertas para productos que no conocíamos pero que ellos intuyen que, por nuestro perfil de consumidor, nos pueden interesar.

Interior de un supermercado en Nueva York (fuente: David Shankbone).

En la red del ejemplo podemos ver que el 98% de los clientes que compraron *Yo, robot* también adquirieron *La fundación*. En cambio, ninguno de los que compraron *Dune* adquirió también *Orgullo y prejuicio*, por lo que no se establece ninguna relación entre estos dos libros. Por eso, si el sistema detecta que un cliente acaba de adquirir *Yo, robot* y ahora está buscando información sobre *La fundación*, en el apartado de recomendaciones ya estará apareciendo *Dune* y *Contacto*, ya que un porcentaje muy significativo de la gente que compró los dos primeros también adquirió estos otros. Con todo ello, el sistema trata de aumentar sus ventas mediante una campaña de marketing individualizada a cada cliente y automatizada para anunciar dos productos que quizás el cliente no sabía que existían, pero como el sistema tiene mucha información de otros compradores previos, ha podido establecer esta red de relaciones causales y la usa para hacer nuevas recomendaciones.

Además, el sistema también sabe que publicitar *Orgullo y prejuicio* a un cliente que está comprando ciencia ficción, que es lo que sucedería si se hiciera una campaña de marketing clásica, es perder el tiempo. En una campaña tradicional de marketing se anunciaría la nueva edición de *Orgullo y prejuicio* dentro del programa temático sobre libros y lectura que se emitiera en el canal cultural a las 23:00 h, por ejemplo. Sin embargo, aunque se habría escogido el programa y la franja horaria donde el telespectador tendría más probabilidad de estar interesado en el producto, seguiría existiendo un gran número de telespectadores aficionados a los libros de ciencia ficción en los que este anuncio no estaría surgiendo ningún efecto, con la pérdida monetaria que esto supone para la empresa anunciante. Para un canal de marketing estático, como la televisión, la radio o los carteles en las calles, es imposible que el anunciante conozca el perfil individualizado de los clientes del momento y, aunque lo supiera, carece de las herramientas necesarias para adaptar el anuncio emitido a cada uno de ellos.

El cerebro del robot: las redes neuronales

La robótica es una de las áreas más complejas de la ingeniería, no sólo por toda la electromecánica y servocontrol que se debe implementar en un simple brazo robotizado, sino por los sofisticados cálculos matemáticos que se requieren para calcular las trayectorias de sus partes móviles. Para ello, en algunos casos el robot cuenta con un cerebro artificial compuesto, del mismo modo que lo están los cerebros de los seres superiores, por redes de neuronas. En este caso, sin embargo, estaremos hablando de neuronas artificiales.

Infografía de una de las neuronas que componen el cerebro humano (fuente: Nicolas P. Rougier).

Los conceptos de «red neuronal» y «neurona artificial» han pasado por diferentes ciclos de euforia-decepción a lo largo de su corta vida. Sus inicios se remontan al algoritmo Threshold Logic Unit (Unidad Lógica Umbral), propuesto por Warren McCulloch y Walter Pitts en la década de 1940 y que tuvo un éxito abrumador. Una neurona artificial es, de hecho, la encapsulación de dicho algoritmo, el cual es definido por los profesionales de la computación como:

$$\text{Entrada}_1 \rightarrow X_1$$
$$\text{Entrada}_2 \rightarrow X_2$$

$$\text{Entrada}_i \rightarrow X_i$$

$$\text{si } \Sigma\ (X_i \cdot \text{Peso}_i) > \text{Umbral},$$
$$\text{entonces Salida} \leftarrow 1$$
$$\text{sino Salida} \leftarrow 0$$

que quiere decir, en lenguaje corriente, que si el estímulo [la suma de productos $(X_i \cdot \text{Peso}_i)$] supera un determinado umbral, entonces, y sólo entonces, se dispara la neurona.

Como puede verse, una neurona es un elemento extremadamente simple, pues sólo implementa unas pocas operaciones aritméticas y una comparación. Este hecho facilitó la implementación de las neuronas artificiales en microchips, de manera que se pudieron implementar redes neuronales completas en hardware a partir de finales de los años noventa. Estos microchips se usan en la actualidad para construir aparatos electrónicos de predicción, como, por ejemplo, los instrumentos capaces de detectar la causa del malestar de un bebé que llora.

El funcionamiento de una neurona artificial es parecido al de una neurona natural y, como se ha podido observar, bastante sencillo. De hecho, la dificultad de las redes neuronales radica principalmente en dos elementos que deben ajustarse y de los cuales dependerá que la red pueda realizar predicciones más o menos correctas: el peso de las diversas entradas y el umbral. El arduo proceso de ajustar estos valores para que, dada una serie de entradas, la neurona produzca la salida deseada es lo que se conoce como «proceso de entrenamiento», o en términos psicológicos, «aprendizaje». La innovación del aprendizaje neuronal fue introducida por Frank Rosenblatt a finales de la década de 1950, y el invento de una neurona que pudiera ajustar los pesos y el umbral fue conocido como «perceptrón».

En términos biológicos, el comportamiento de una neurona natural es prácticamente el mismo: cada neurona tiene un conjunto de entradas por las que percibe señales eléctricas provenientes de otras neuronas, lo que se conoce como «conexiones sinápticas», y a partir de ellas evalúa si esos estímulos superan un umbral de sensibilidad, siempre teniendo en cuenta que hay conexiones sinápticas que son más importantes que otras (los pesos de los que se hablaba antes). Si se supera este umbral de sensibilidad, se propaga una señal eléctrica a través del axón, o lo que sería su equivalente en una neurona artificial, la salida.

Mediante esta relativa simplicidad, el perceptrón se planteó como una útil herramienta predictiva: dada una muestra, predice si es de una clase (0) o de otra (1). Un ejemplo clásico es el problema del género botánico *Iris*, en el que se tienen muestras de tres especies: *Iris setosa*, *Iris versicolor* e *Iris virginica*. Cada muestra recogida se define por cuatro parámetros: la longitud de sus pétalos, la amplitud de los mismos, la longitud de sus sépalos y la amplitud de éstos. El objetivo es que, dada una nueva muestra, la herramienta nos diga a qué especie pertenece. En este caso, usaremos tres perceptrones, cada uno especializado en la detección de una sola de las tres especies, de tal manera que si la nueva muestra es de la especie *Iris setosa*, sólo uno de los perceptrones debería retornar 1, y los otros dos, 0.

Iris setosa, I. versicolor e I. virginica, *en este orden. Según la forma y dimensión de los pétalos y sépalos, el sistema es capaz de categorizar nuevas flores.*

En estos momentos, es posible que el lector esté pensando que por qué no se usan herramientas de tipo estadístico para resolver el problema del *Iris*. En efecto, este problema es tan simple que también se podría resolver mediante herramientas estadísticas clásicas, como el análisis de componentes principales. Sin embargo, nótese que el perceptrón y las herramientas estadísticas nos aportan dos maneras de razonar muy distintas, y que tal vez la más similar al razonamiento natural sea la del perceptrón.

En el caso de utilizar herramientas estadísticas, en el problema del *Iris* hubiéramos obtenido reglas del tipo: «si los pétalos están entre tal y tal longitud, y su anchura está entre ésta y esta otra, es probable que esta muestra sea de la especie X». Mientras que la manera de razonar de un perceptrón es la siguiente: «si los pétalos están entre tal y tal longitud, y su anchura está entre ésta y esta otra, es probable que esta muestra sea de la especie X, a no ser que tenga unos sépalos tan cortos que entonces ya no me importa cómo sean los pétalos porque puedo decir que es de la especie Y».

Es decir, mediante el sistema de ponderar el valor de las entradas para la toma de decisiones hay alguna información que puede pesar mucho menos que otra, pero que, si se llega a un valor extremo, esta entrada antes poco importante pasa a convertirse en muy relevante para la toma de la decisión.

Las neuronas se agrupan

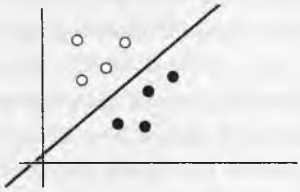
A pesar de la gran innovación que supuso el perceptrón y del amplio rango de aplicabilidad que se le preveía, pronto se descubrió que no podía operar en un determinado grupo de problemas llamados «no-separables linealmente». Por desgracia, la mayoría de problemas de la vida real son de este tipo. Por eso, en la década de 1980 surgieron numerosos críticos a las redes neuronales en un debate que a menudo sobrepasaba lo científico y entraba a descalificar personalmente a los defensores del perceptrón.

Este hecho coincidió, para mayor decepción y frustración de los científicos que investigaban en estas áreas, con la época oscura de la inteligencia artificial, etapa que se vio marcada por un más que significativo descenso de los fondos dedicados a ella, tanto en Estados Unidos como en Europa. En primer lugar, la sociedad se dio cuenta de que la idea que se había vendido en películas como *2001: Una odisea en el espacio* no iba a ser cierta en un larguísimo lapso de tiempo; en segundo lugar, las agencias gubernamentales estadounidenses, que habían

Durante años se pensó que la creación de ordenadores superinteligentes como HAL 9000, de 2001: Una odisea en el espacio, *era una posibilidad muy real. Pronto llegó la decepción.*

LA NO-SEPARABILIDAD LINEAL

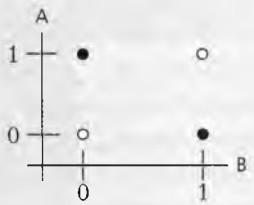
Si consideramos una situación en la que las muestras puedan ser de dos categorías y cada una de ellas esté descrita por dos descriptores (por tanto, dos entradas), podríamos dibujar un gráfico como el siguiente, con ocho muestras recogidas.



En él, los círculos blancos representan las muestras de la categoría A, y los negros, las de la categoría B. Como puede apreciarse, es fácil trazar una línea que separe ambas categorías, y eso es justamente lo que hace un perceptrón cuando se ajustan el umbral y los pesos de cada entrada. Sin embargo, ¿qué pasa si analizamos el problema sintético de la X-OR? X-OR es una operación lógica (corresponde al «o» exclusivo) que cumple con la siguiente relación:

Entradas	Salida
00	0
01	1
10	1
11	0

Ahora el gráfico queda así:

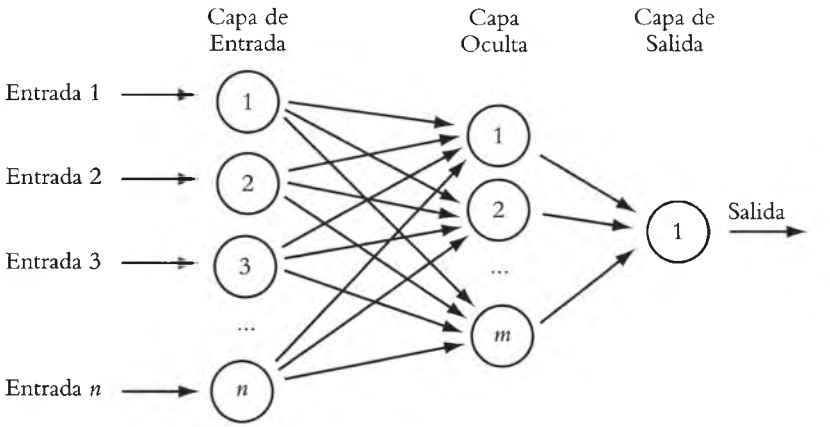


En este caso no es posible dibujar ninguna recta que separe los círculos blancos de los negros y, por tanto, estamos ante un problema no-separable linealmente. Un perceptrón no podría ser entrenado correctamente en la solución de un problema lógico tan simple como el de X-OR.

depositado grandes esperanzas en la inteligencia artificial para decantar a su favor la Guerra Fría, sufrieron notables fracasos, como el de la traducción automática, que era de capital importancia para la interpretación de documentos técnicos rusos. A pesar de la enorme reducción de fondos que supuso descubrir la ineficacia de los perceptrones en los problemas no-separables linealmente, la investigación en el tema continuó, si bien a una velocidad mucho menor y algunas veces en secreto, con el fin de que sus investigadores no fueran ridiculizados por la gran mayoría de detractores. Pero ¿se podía solventar realmente el problema de la no-separabilidad lineal?

La respuesta vino a finales de los años ochenta y era tan obvia y natural que, de hecho, no se comprende cómo los investigadores no se habían percatado antes. La propia naturaleza ya había descubierto la respuesta millones de años atrás: la solución estaba en conectar diversos perceptrones entre sí para formar lo que se conoce como «redes neuronales».

En la figura siguiente se muestra una red neuronal formada por tres capas neuronales: la primera es la de entrada, la segunda es la oculta y la última, la de salida. A esta red neuronal se la denomina *«feed forward»* (algo así como «alimentado hacia delante»), ya que el flujo de los datos siempre es de izquierda a derecha y no se forman ciclos entre las conexiones sinápticas.



Pero una red neuronal puede ser tan compleja como se desee, con tantas capas ocultas como el programador considere oportuno y, además, con conexiones que pueden ir desde delante hacia atrás para simular una especie de memoria. En este

sentido, se han llegado a construir redes neuronales con 300.000 neuronas, que es el mismo número de neuronas que contiene el sistema nervioso de una lombriz de tierra.

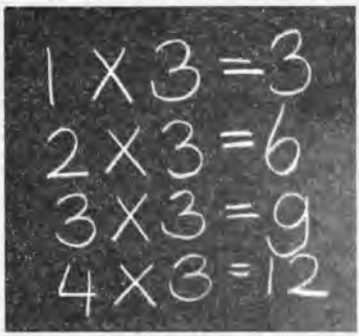
En una red neuronal el aprendizaje se complica, y por ello los ingenieros han ideado un gran número de métodos de aprendizaje. Uno de los más simples es el método de la retropropagación, o en inglés *back-propagation*, que además da nombre a las redes neuronales que hacen uso de él. Este método consiste en minimizar el error de salida de la red neuronal a base de ajustar de derecha a izquierda los pesos de entrada de las conexiones sinápticas de las neuronas siguiendo el método del descenso de gradiente. Es decir, que primero se les dan valores aleatorios a los pesos de todas las conexiones de la red y a continuación se introduce una muestra con el valor conocido que se debe predecir; por ello se dice que es una muestra «de entre-

LA TRAMPA DEL SOBREENTRENAMIENTO

Un sistema de predicción basado en el aprendizaje artificial deduce sus predicciones de la generalización que es capaz de hacer a partir de casos pasados. Por tanto, cuando el sistema no es capaz de generalizar, éste pierde su utilidad.

Cuando el proceso de entrenamiento se repite demasiadas veces llega un punto en el que el ajuste es tan exacto y está tan adaptado a las muestras de entrenamiento que el sistema, como las ha memorizado, ya no basa sus predicciones en una generalización sino en la memorización. Cuando esto sucede, el sistema ya sólo es capaz de realizar predicciones correctas en el caso de que se le introduzcan muestras del conjunto de entrenamiento, y siempre que se le introduzca una muestra distinta para que realice una predicción, ésta será incorrecta. Entonces se dice que el sistema está «sobreentrenado».

De alguna manera, es lo mismo que le ocurriría a un niño que, en vez de aprender a multiplicar, memorizase solamente las tablas. Si se le preguntara por una de las operaciones que ha memorizado, contestaría correctamente sin dudarlo, pero no sucedería lo mismo si se le preguntase por una nueva multiplicación que no forma parte de las tablas.

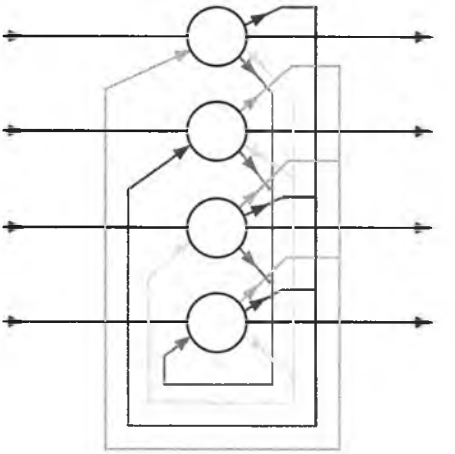


Las tablas de multiplicar son un buen ejemplo de aprendizaje por memorización.

namiento». Como es de esperar, el resultado que retornan las neuronas de salida es un valor aleatorio. A partir de aquí, comenzando por las neuronas más cercanas a la salida y terminando por las de la entrada, se empiezan a ajustar los valores de los pesos de las conexiones con el objetivo de que el valor de la neurona de salida se acerque al valor real conocido.

Este procedimiento se repite centenares o miles de veces con todas las muestras de entrenamiento. Cuando se ha completado el proceso con todas ellas, se dice que ha pasado una época. A continuación, se puede repetir el proceso otra época entera con las mismas muestras. Un proceso de aprendizaje normal consta de unas pocas decenas de épocas. Obsérvese que este proceso es similar al proceso psicológico de aprendizaje, en el que se aprende por repetidas observaciones de los mismos datos.

La arquitectura de las redes neuronales y los métodos de aprendizaje fueron sofisticándose a lo largo de los años, y poco a poco fueron surgiendo un gran número de tipos de redes neuronales diferentes para aplicaciones muy diversas de la vida real. Unas de las más usadas en la actualidad son las de Hopfield, que implementan un tipo de dispositivo de memoria llamado «memoria asociativa».



Esquema de una red neuronal de Hopfield.

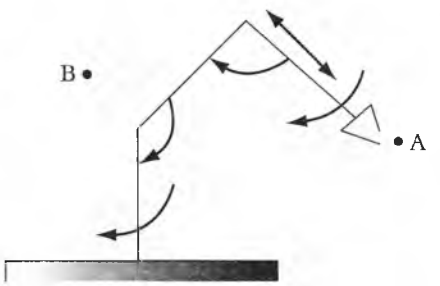
En una memoria asociativa, la información está ordenada según el contenido y, por tanto, para acceder a ella hay que indicar a qué contenido queremos acceder, en vez de dar una posición física-electrónica, como se hace en un disco duro o en la memoria RAM de un ordenador.

Otro tipo de redes neuronales muy usadas en la actualidad son las redes o mapas autoorganizativos o mapas de Kohonen. En este caso, la red neuronal incorpora una innovación: el aprendizaje no es supervisado, sino que la propia red va autoaprendiendo de sus errores a medida que va trabajando.

Y el cerebro funciona

Existe una disciplina de la física, la cinemática inversa, que se encarga de calcular los movimientos necesarios que se deben desencadenar para trasladar un objeto desde un punto del espacio A hasta otro punto B. Estos cálculos se complican de forma exponencial, en cuanto al número de operaciones matriciales que se deben resolver, a medida que se introducen grados de libertad en el sistema.

Por ejemplo, dado un brazo robotizado con cuatro ángulos de torsión y una extensión retráctil como el que aparece en la figura siguiente, si se quisieran resolver por la vía clásica las ecuaciones matriciales de cinemática inversa, un superordenador podría tardar horas en realizar todos los cálculos necesarios para saber con qué magnitud y en qué sentido debemos desplazar cada grado de libertad para llevar la herramienta situada en el extremo del brazo desde A (punto inicial) hasta B (punto final).



Por ello es imposible mantener sistemas robotizados que modifiquen sus trayectorias en tiempo real basándose en la resolución clásica de matrices. En el caso de aquellos robots que ejercen sistemáticamente las mismas tareas repetitivas (piénsese, por ejemplo, en los robots de una cadena de montaje de una fábrica de automóviles) se pueden calcular y programar *a priori* todos los movimientos, paso a paso, que deben ejecutar los motores de cada ángulo de torsión o brazo retráctil del robot. Sin embargo, si se desea implementar un brazo robotizado capaz de actuar autónoma-

mente y con capacidad para coordinar sus acciones en función de una situación concreta (piénsese ahora en los robots ubicados en astronaves, en los empleados en cirugía o, simplemente, en los primeros robots experimentales de uso doméstico), se necesitan sistemas más innovadores que permitan a los elementos de proceso del robot calcular de manera rápida cómo se tiene que mover para cumplir con su cometido.

Para ello, uno de los sistemas de control motor más eficientes que existen en la actualidad son las redes neuronales de retropropagación. En el caso del robot, la red neuronal entrenada para su control tendrá tantas salidas como grados de libertad disponga el robot, y cada una de ellas indicará la magnitud y el sentido con el que debe desplazarse cada motor para ir desde el punto inicial hasta el final.

Sin embargo, el gran inconveniente de esta aproximación es que la red neuronal debe pasar por un largo proceso de aprendizaje, innecesario en una aproximación clásica. En cierto modo, se podría ver como el proceso de aprendizaje de un ser humano, que de pequeño aprende a caminar a base de prueba y error, pero que, una vez que ha aprendido a andar, no hace falta que esté solventando a cada paso las complejas ecuaciones físicas que resuelven la cinemática de sus piernas para desplazarse y no perder el equilibrio.

Por ello, el entrenamiento habitual que se usa en estos casos es el de mostrar a la red neuronal de retropropagación decenas de miles de veces, una y otra vez, centenares de miles de trayectorias posibles y enseñarle cómo se mueven en cada caso los distintos motores para pasar del punto inicial al final.

Una vez que la red está adecuadamente entrenada, se dice que ha aprendido el mapa sensoriomotor. Esto permite que el centro de procesado de un robot pueda resolver con precisión milimétrica la cinemática inversa del robot en escasos milisegundos.

Y el cerebro se complica

Los buenos resultados obtenidos por las redes neuronales provocaron que, ya en el siglo XXI, éstas se estandarizaran como la herramienta modelo que había que usar ante muchos problemas. Sin embargo, tenían aún grandes puntos débiles.

El primero de ellos era el sobreentrenamiento; es bastante fácil caer en él durante el entrenamiento de una red neuronal. El segundo era el gran número de parámetros que deben fijarse «a mano» y de manera arbitraria antes de pasar a la etapa de aprendizaje de la red neuronal, y el problema de ajustar todos esos importantes

factores «a mano» radica en que no existe ningún manual ni ninguna metodología para hacerlo, con lo cual se deben invertir grandes recursos humanos y técnicos (la mayoría de las veces mediante el antiguo y detestable método de prueba y error) hasta ajustar todas estas características. Un tercer problema, ya más filosófico que práctico, es la incomprensión del razonamiento de una red neuronal una vez está entrenada. A este hecho no se le dio importancia hasta que las redes neuronales empezaron a introducirse plenamente en la resolución de problemas reales. Si, por ejemplo, se usa una red neuronal para controlar el sistema de frenado ABS de un coche, es lógico que los ingenieros quieran asegurarse de entender, hasta el último detalle, cuál es el razonamiento de la red neuronal, para así poder asegurar que los frenos no van a fallar ante ninguna de las miles de situaciones diferentes de frenado en que se pueden ver implicados.

La red neuronal realiza predicciones, pero no se sabe qué razonamiento sigue para obtenerlas. Hay quien la compara con una bola de cristal.

Por estos importantes motivos, diversos teóricos de la computación trabajaron intensamente desde finales de los años noventa para diseñar nuevos métodos computacionales que solventaran o mitigaran estos efectos. La solución final llegó a principios del siglo XXI de la mano de Vladimir Vapnik y de su equipo, de la famosa empresa de telecomunicaciones y equipos electrónicos AT&T Bell Labs. Vapnik ideó las máquinas de soporte vectorial (Support Vector Machine, SVM por sus siglas en inglés), que se basan principalmente en introducir nuevas dimensiones artificiales a un problema no-separable linealmente para que, gracias a las nuevas dimensiones artificiales introducidas, ahora sí lo sea.

Las SVM logran superar la mayoría de los inconvenientes que aparecían en las redes neuronales (sobre todo sobreentrenamiento, fijación de parámetros iniciales e incomprensibilidad de su razonamiento), por lo que en la actualidad las han sustituido en prácticamente todos los ámbitos de la computación. Sin embargo, todavía

se utilizan en algunas aplicaciones industriales, como en el campo de la robótica, debido a la simplicidad de su implementación en hardware.

¿Son necesarios los exámenes?

¿Se puede aplicar el aprendizaje artificial al aprendizaje natural? Como se ha ido viendo, el aprendizaje artificial se puede aplicar a todos los ámbitos de la ciencia y de la tecnología, pero ¿es posible ir más allá y aplicarlo a las ciencias sociales, y en particular a la educación? ¿Cómo determina un profesor el nivel de conocimiento alcanzado por sus alumnos? ¿Se pueden automatizar algunos de los criterios subjetivos que profesores y maestros aplican a menudo para evaluar a sus alumnos? Incluso, ¿se puede predecir el rendimiento de un alumno sin tener que examinarlo? Todo ello se puede realizar mediante una técnica tan sencilla como los árboles de decisión.

¿Desaparecerá esta imagen de las aulas del futuro? Seguro que muchos estudiantes estarían felices de que así fuera...

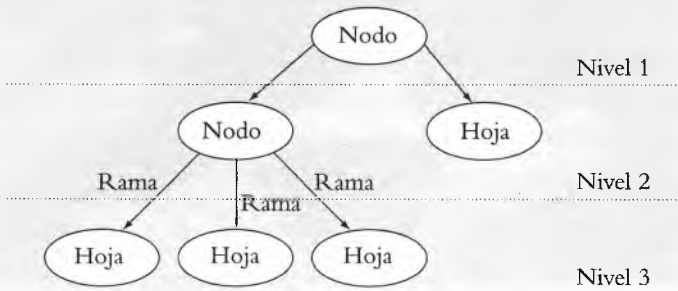
Los árboles de decisión son una herramienta muy simple, pero también muy efectiva, para el reconocimiento de patrones. Un árbol de decisión trata de buscar cuáles son las variables más decisivas o discriminantes que decantan una muestra

hacia una clase u otra. Veamos un ejemplo: estamos ante un problema de predicción de notas de los alumnos de un determinado instituto y tenemos los siguientes datos de entrenamiento:

Nota del curso anterior	Asistencia a clase	Ejercicios entregados	Objetivo: ¿Aprobado o suspenso?
Mayor que la media	100%	45%	Suspenso
Mayor que la media	100%	100%	Aprobado
Mayor que la media	90%	100%	Aprobado
Menor que la media	85%	30%	Suspenso
Menor que la media	100%	80%	Aprobado
Menor que la media	99%	100%	Suspenso
Menor que la media	100%	55%	Suspenso

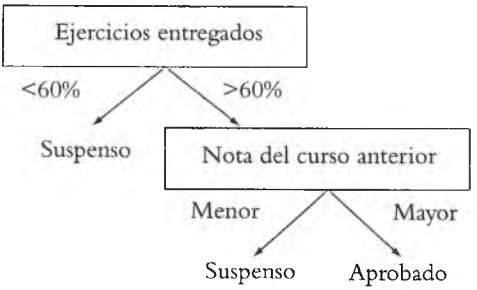
EL ÁRBOL DE LA INFORMACIÓN

El árbol es un tipo de estructura organizativa de la información muy utilizada en ingeniería, ya que permite relacionar los datos de una manera jerarquizada. El árbol, como estructura de datos, utiliza una nomenclatura propia, que es interesante conocer:



Cada uno de los datos de un árbol se denomina «nodo», y estos nodos, que representan una unidad de información, se dividen en los distintos niveles y se relacionan entre ellos mediante ramas. Un nodo relacionado con otro de distinto nivel será su padre si su nivel es superior, o su hijo, en caso contrario. Finalmente, los nodos que no tienen hijos son denominados «hojas».

Un árbol de decisión que modela bien estos datos puede ser el siguiente:



En este caso, la asistencia a clase no es una variable discriminante, ya que no forma parte de los nodos del árbol. Existen distintas metodologías para determinar si la variable de un modelo es discriminante. Una de las más utilizadas es la que se basa en la entropía de Shannon. Según este método, en cada nivel del árbol se analiza cuál es la variable que genera una menor entropía, y ésta es la que se selecciona para discriminar en ese nivel del árbol. Vamos a verlo con más detalle.

La entropía de Shannon, *S*, viene definida por la siguiente fórmula:

$$S = -\sum_{i=0}^{n} n_i \cdot \ln(n_i).$$

Veámosla en acción usando el ejemplo de los exámenes. En el primer nivel del árbol deberemos analizar la entropía generada por cada variable. La primera de ellas es la variable «nota del curso anterior». Si separamos las muestras por esta variable nos quedan dos subconjuntos de muestras, uno con

$$S_{\text{Nota del curso anterior menor que la media}} = -0,75 \cdot \ln(0,75) - 0,25 \cdot \ln(0,25) = 0,56,$$

ya que de entre los alumnos que el año pasado tuvieron una nota menor que la muestra hay un 75% de suspensos y un 25% de aprobados, y otro con

$$S_{\text{Nota del curso anterior mayor que la media}} = -0,33 \cdot \ln(0,33) - 0,67 \cdot \ln(0,67) = 0,64,$$

ya que un tercio de los alumnos que el año pasado tuvieron una nota mayor que la media han suspendido y dos tercios han aprobado.

Esta operación se repite para cada variable. Y la siguiente es la «asistencia a clase», en la que, para simplificar, discerniremos entre mayor que el 95% y menor que el 95%. En este caso,

$$S_{\text{Asistencia a clase mayor que el 95\%}} = -0,6 \cdot \ln(0,6) - 0,4 \cdot \ln(0,4) = 0,67;$$
$$S_{\text{Asistencia a clase menor que el 95\%}} = -0,5 \cdot \ln(0,5) - 0,5 \cdot \ln(0,5) = 0,69.$$

Finalmente, analizaremos la variable «ejercicios entregados», en la que, de nuevo para simplificar, dividiremos en mayor que el 60% y menor que el 60%. Así:

$$S_{\text{Ejercicios entregados mayor que 60\%}} = -0,75 \cdot \ln(0,75) - 0,25 \cdot \ln(0,25) = 0,56$$

y

$$S_{\text{Ejercicios entregados menor que 60\%}} = -1 \cdot \ln(1) = 0.$$

Por tanto, la variable que mejor discrimina es esta última, puesto que las entropías de los subconjuntos que genera son de 0,56 y 0.

En este caso, las muestras de entrenamiento que caen en el lado de «menos del 60% de los ejercicios entregados» son todo suspensos, por tanto, ya no nos tenemos que preocupar de esta rama del árbol. Sin embargo, la otra rama contiene tanto suspensos como aprobados; por consiguiente, debemos proseguir el análisis de manera recursiva, eso sí, sin tener en cuenta las muestras que ya han sido discriminadas.

Ahora sólo nos quedan dos posibles variables de decisión: «nota del curso anterior» y «asistencia a clase». Las entropías de Shannon de los grupos generados mediante la primera variable discriminante son las siguientes:

$$S_{\text{Nota del curso anterior menor que la media}} = -0,5 \cdot \ln(0,5) - 0,5 \cdot \ln(0,5) = 0,69;$$
$$S_{\text{Nota del curso anterior mayor que la media}} = -1 \cdot \ln(1) = 0,$$

mientras que si se analiza el comportamiento en «asistencia a clase», tendremos:

$$S_{\text{Asistencia a clase mayor que el 95\%}} = -0,33 \cdot \ln(0,33) - 0,67 \cdot \ln(0,67) = 0,64;$$
$$S_{\text{Asistencia a clase mayor que el 95\%}} = -1 \cdot \ln(1) = 0.$$

Por tanto, se escoge la variable discriminante de «asistencia a clase», ya que presenta unas entropías menores.

El método de construcción de los árboles de decisión y, por tanto, el método que hace que los árboles aprendan, es sencillo y elegante, pero tiene dos grandes inconvenientes. El primero es que en problemas con un gran número de variables de decisión resulta muy lento, y el segundo, y más grave, es que puede caer fácilmente en óptimos locales, es decir, como nunca se analiza el árbol en su globalidad,

sino nivel a nivel, es posible que una determinada variable de decisión minimice la entropía en un determinado nivel y se escoja, pero en cambio si se hubiera escogido otra, el árbol clasificaría mejor a escala global.

Un «truco» que se utiliza habitualmente para mejorar los índices de acierto de los árboles de decisión es usar «bosques», es decir, entrenar diversos árboles, cada uno con un método diferente, y obtener la predicción final del consenso de la predicción de cada uno de los diversos árboles que componen el bosque.

Siguiendo esta filosofía, la metodología más frecuente de entrenar un bosque es construir árboles de decisión mediante la selección aleatoria de variables de decisión; es decir, si queremos entrenar unos 100 árboles que compondrán nuestro bosque, para cada árbol se seleccionan unas 5 variables de decisión aleatorias, y se entrena el árbol sólo con aquellas 5 variables. Esta aproximación se conoce con el poético nombre de «*random forest*» («bosque aleatorizado»).

Capítulo 4

Planificación y razonamiento automático

Los sucesos siguientes podrían ocurrir un día cualquiera:

14:32 h: Un camión con exceso de velocidad vuelca en una carretera secundaria. El conductor recibe un fuerte golpe en la cabeza.

14:53 h: Acuden al lugar del accidente una ambulancia y los bomberos, y en pocos minutos consiguen evacuar al conductor, inconsciente y con una grave fractura craneal.

15:09 h: La ambulancia llega al hospital, donde el servicio de urgencias decreta la muerte cerebral del conductor.

15:28 h: Se identifica al sujeto y se avisa a su familia.

16:31 h: Una vez en el hospital, un equipo de psicólogos se pone en contacto con la familia del fallecido para darle soporte emocional y obtener su consentimiento para la donación de los órganos no afectados por el traumatismo.

16:36 h: Después de un breve debate, la familia acepta donar los riñones del familiar fallecido (en adelante, el donante).

16:48 h: Un equipo quirúrgico empieza los trabajos de extracción de los riñones y la exploración médica de éstos. En paralelo, los responsables administrativos del hospital trabajan en los trámites burocráticos para obtener la autorización legal.

17:24 h: Una vez concluida la extracción, se introducen en un sistema informático los datos biológicos del donante y las características de los órganos.

Así comienza un trasplante de órganos.

Cómo se gestiona un trasplante

17:24 h: De manera inmediata, el sistema informático determina quiénes serán los dos receptores de los riñones, les avisa y asigna y planifica los recursos

logísticos necesarios para el transporte. En un caso hay que preparar una ambulancia para llevar uno de los riñones al hospital de una ciudad vecina (a 30 km de distancia), y en el otro es una avioneta de transporte médico la que llevará el riñón a una ciudad que está a 450 km de distancia y pertenece a otro sistema autonómico de salud. El transporte del segundo órgano desde el hospital del donante hasta el aeropuerto más cercano se realiza mediante un helicóptero, que el mismo sistema informático asigna de manera automática. En paralelo, este sistema está realizando también gran parte de los trámites legales que requieren los dos sistemas sanitarios, el del donante y el del receptor de la región vecina.

18:10 h: Empieza el primer trasplante en el hospital de la ciudad vecina.

19:03 h: Comienza el segundo trasplante en la ciudad situada a 450 km.

21:00 h: Ambos receptores están recibiendo el tratamiento postoperatorio e inmunosupresor adecuado, y los dos evolucionan favorablemente.

¿Por qué el sistema español de trasplante de órganos está considerado como uno de los mejores del mundo? ¿Qué tiene este sistema que no tienen los de otros países con una posición tecnológica y científica más avanzada que la española, que hace que incluso la Comisión Europea esté estudiando la extensión de este modelo a todos los países de la Unión? Como el lector tal vez ya habrá intuido, el sistema de trasplantes español está basado en un potente sistema de inteligencia artificial distribuido por toda la red hospitalaria del país, que no sólo contempla y tiene en cuenta las necesidades y características de cada receptor y todos los detalles logísticos, sino también la compleja y fragmentada normativa de trasplantes del país.

Dicho sistema inteligente está basado en un sistema multiagente, lo que significa que está construido a partir de muchos sistemas informáticos, relativamente sencillos pero muy especializados en su función, los cuales conforman una potente inteligencia colectiva que lo convierte en el mejor modelo mundial de trasplantes. Un sistema de coordinación de trasplantes normalmente tiene una estructuración multinivel, por ejemplo, a escala nacional, zonal, regional y hospitalaria, y además, en el ámbito hospitalario, los datos de los receptores se pueden mantener distribuidos por la red de hospitales o agrupados en un *data warehouse* («almacén de datos»). Por todo ello, existe una gran cantidad de agentes inteligentes que gestionan la información de los receptores y que son consultados continuamente por otros agentes inteligentes que se activan cada vez que hay un do-

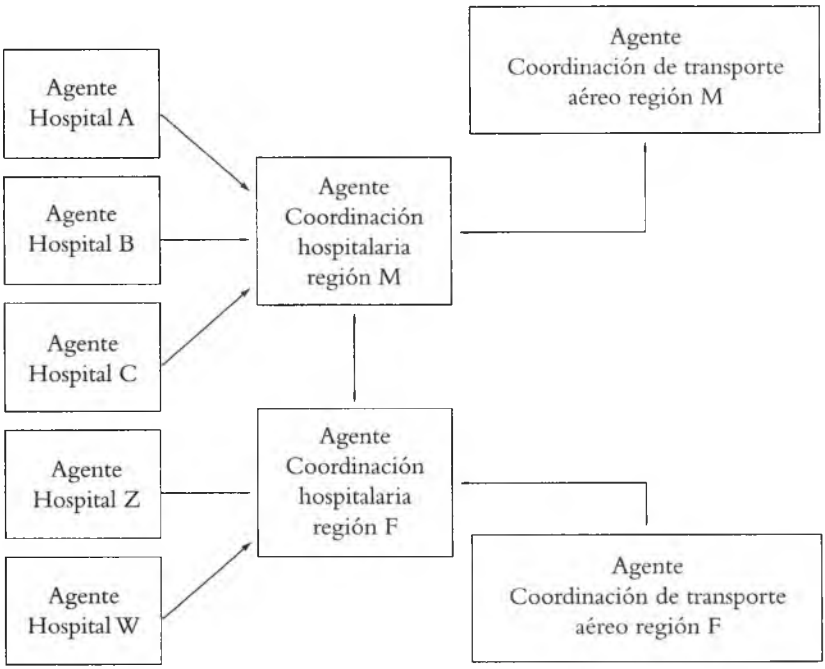
EL *PIGEON RANKING*

La celebérrima empresa Google presume de una gran reputación en el mundo de la computación y de la inteligencia artificial. De hecho, su santo grial es el algoritmo que usa para confeccionar los rankings de las búsquedas que los internautas de todo el mundo realizan millones de veces por segundo. Tanto interés despierta ese algoritmo y tantas presiones recibía Google para publicarlo que la mañana del 1 de abril de 2002 la empresa decidió poner un link desde su página principal donde afirmaba que explicaba su algoritmo de ranking. El algoritmo en cuestión se denominaba *pigeon ranking* («ranking de palomas»), y su extraño nombre no era en absoluto un capricho. Contaba el artículo que Google disponía de unas instalaciones llenas de PC (siglas de *pigeon cluster* en inglés, es decir, «agrupamiento de palomas») y que cada paloma disponía de una pantalla y un teclado. La descripción del algoritmo decía que cada vez que un usuario realiza una búsqueda en Google, cada una de las webs que coincide con dicha búsqueda es mostrada a una de las palomas, que acto seguido empieza a picotear en su teclado. A continuación, las webs son ordenadas según el número de picotazos que ha dado su paloma correspondiente.

En el mismo artículo, Google explicaba cómo manipulaban las palomas, cómo vivían y cómo eran reclutadas. También se decía que habían probado con otro tipo de aves, como las gallinas o distintas especies de aves de presa, pero que las palomas se habían revelado como las más inteligentes y adecuadas para la tarea. El artículo incluso se atrevía a afirmar que, si bien era cierto que ninguna paloma había llegado aún a ser miembro del Tribunal Constitucional, en cambio sí que se había probado su eficacia como controladores aéreos y árbitros de fútbol.

Muchos ingenieros y competidores no daban crédito a lo que estaban leyendo, hasta que la gente empezó a caer en la cuenta de que el día 1 de abril en Estados Unidos y en muchos otros países es el *Aprils Fools*, algo parecido al Día de los Santos Inocentes en algunos países de habla hispana, en que la gente se suele gastar inocentadas durante toda la jornada.

nante. Otros agentes del sistema se encargan de aspectos diversos, como la planificación y la asignación de recursos logísticos para el transporte crítico de los órganos o la gestión de los trámites administrativos que requieren los diversos sistemas regionales de salud.



Ejemplo simplificado de una red de agentes encargada de la coordinación de trasplantes de órganos.

Trabajar con una arquitectura multiagente inteligente como ésta reporta muchas ventajas, como el hecho de que el sistema es redundante a fallos, puesto que si un agente o conjunto de agentes falla, se puede autorregular y asignar otros agentes para la realización de las tareas que deben desarrollarse. Otra gran ventaja, y que se aprecia de forma muy clara en el caso de los trasplantes, es que a base de implementar agentes relativamente sencillos pero muy especializados, se puede construir un sistema inteligente capaz de resolver en pocos segundos diversas tareas complejas, interdisciplinares y críticas a la vez.

PROGRAMACIÓN ORIENTADA A AGENTES

La programación de computadoras es un campo que evoluciona rápidamente. En la actualidad existen cinco grandes familias de lenguajes de programación, y aquella que está orientada a objetos es la más usada entre los programadores actuales. Se trata de un tipo de programación en la que todo está representado mediante unidades de información llamadas «objetos». Éstos cuentan con una serie de atributos donde se guarda la información sobre sí mismos y son capaces de realizar algunas operaciones sobre dicha información. En la orientación a objetos se requiere la intervención de coordinadores que posean la inteligencia y que vayan solicitando a los objetos que realicen una u otra tarea, pero siempre asumiendo que el objeto es básicamente un elemento poco inteligente que espera que le ordenen qué tiene que hacer. Sin embargo, últimamente está emergiendo con fuerza una evolución de estas programaciones: la orientada a agentes. En ella, los objetos «tontos» pasan a ser agentes con mucha más inteligencia y autonomía, de modo que la labor del coordinador ya no es tan crítica.

Planificar, ésa es la palabra

Planificar una serie de recursos, más o menos numerosos, para que una determinada tarea se lleve a cabo con éxito puede ser una actividad extremadamente compleja, hasta para un experimentado humano. Además, la planificación aparece en todos los ámbitos de la vida real, desde tareas poco críticas, como la planificación de profesores, asignaturas, grupos de alumnos, aulas, laboratorios o dispositivos audiovisuales de los que se disponga en un instituto de secundaria, hasta la crítica planificación de los recursos en la extinción de un incendio forestal o en una emergencia ante cualquier otra catástrofe natural.

Por otra parte, el razonamiento automático es una actividad extremadamente sencilla para un humano, pero realmente compleja para una máquina. De hecho, «razonar» es lo que en gran medida caracteriza a un ser humano y las claves de su razonamiento aún no están claramente desveladas por los neurobiólogos y expertos en la materia. Con el fin de imitar ese razonamiento, los ingenieros han inventado algunas técnicas muy interesantes, aplicables, por ejemplo, a los incendios forestales.

Hoy en día, son ya varias las administraciones que basan sus protocolos de extinción de incendios forestales en sistemas planificadores basados en inteligencia artificial. Habitualmente, cuando se declara un incendio forestal de tamaño medio, un técnico de extinción tarda entre una hora y una hora y media en diseñar un plan

de extinción; dicho plan detalla el proceso que deben seguir los medios disponibles en ese momento en función de una serie de características del terreno, meteorológicas, etc. Sin embargo, uno de los problemas a los que se enfrentan normalmente estos técnicos es que las condiciones son cambiantes y, además, pueden verse alteradas con una frecuencia mayor de lo que se tarda en rehacer el plan de extinción. Por ello, muchas administraciones están tratando de implantar sistemas automatizados que puedan construir estos planes de extinción en cuestión de segundos. De esta manera, el sistema recoge parámetros como la orografía del terreno, las circunstancias meteorológicas, los accesos disponibles a la zona afectada por el incendio, la disponibilidad de medios aéreos o terrestres, o la coordinación y comunicación con diferentes unidades y centros de control, y a partir de ahí diseña un plan que a continuación revisa el experto humano.

Un incendio forestal exige la coordinación de numerosos recursos humanos y materiales.

Podría suceder que, por ejemplo, en un momento dado una unidad terrestre quedara libre y al sistema se le plantearan dos opciones: desplazarla a una zona donde el incendio es muy activo o enviarla a que apague las llamas a otra zona más cercana y menos peligrosa. ¿Cómo puede decidir el sistema cuál de las dos opciones es la mejor? Lógicamente, el objetivo es apagar el incendio y, por tanto, parece más plausible que la unidad se dirija a la zona donde el fuego es más virulento; pero, por

otro lado, puede tardar horas en llegar allí, mientras que a pocos minutos tiene una zona donde las llamas son menos peligrosas y se podrían apagar con cierta facilidad, de modo que se atajaría el incendio en aquel lugar. La pregunta es: ¿cómo se cuantifica de una manera clara y objetiva, tal y como requeriría un sistema de planificación no inteligente clásico, el beneficio de apagar el fuego en una zona atendiendo a la distancia que se ha de recorrer y al tiempo transcurrido? Si esta unidad terrestre se multiplica por las decenas de ellas que intervienen en la extinción de un incendio, se le suman las unidades aéreas y se añaden nuevas variables como la intensidad del viento y su previsión, lluvias, zonas habitadas, zonas de alto valor ecológico, etc., podemos empezar a comprender hasta qué punto se requiere disponer de un sistema inteligente que sea capaz de tomar decisiones teniendo en cuenta todos los detalles y obedeciendo a parámetros difusos.

LA LÓGICA DIFUSA

La lógica difusa es un tipo de lógica matemática que trata de aproximar los métodos y operaciones lógicas al proceder humano y natural. Normalmente, en un entorno real las cosas nunca son blancas o negras; en cambio, en las lógicas clásicas, como puede ser la lógica booleana, los valores sólo pueden ser ciertos o falsos, con lo que se nos obliga a posicionarnos siempre en los extremos.

Por ejemplo, si yo pregunto si el portero de un equipo de la primera división de Kazajistán es bueno o malo, la respuestan será «depende», ya que si se compara con la élite del fútbol mundial, lo más seguro es que no sea lo bastante bueno, pero, en cambio, si se compara con el portero del equipo de fútbol de mi barrio, probablemente sea un portero excelente. Por eso, las variables de la lógica difusa no contienen valores de «verdadero» o «falso», sino un valor real, normalmente entre 0 y 1, simbolizando que si el valor es «1» es «verdadero» y si es «0» es «falso». Siguiendo con el ejemplo, si cualificamos el 0 como una nulidad absoluta a la hora de parar cualquier tipo de chut y el 1 como la excelencia del mejor portero del mundo, seguramente el portero kazajo se encontrará en un respetable 0,73.

Para abordar este tipo de problemas, lo habitual es usar técnicas de búsqueda clásicas dentro de la inteligencia artificial, como pueden ser el *back-tracking* («vuelta atrás») o el *branch-and-bound* («ramificación y poda»). Ambas técnicas tienen un funcionamiento análogo: básicamente, desplegar un árbol de combinaciones y recorrerlo hasta encontrar la mejor alternativa. En un problema de planificación, «des-

plegar un árbol de combinaciones» no significa más que generar el árbol de todas las planificaciones posibles (recuérdense los conceptos explicados en el primer capítulo sobre cómo un algoritmo inteligente puede resolver el problema del ajedrez) e ir «podando» de manera inteligente aquellas ramas del árbol que nos lleven a planificaciones no posibles, que violen alguna restricción o que se prevea que no conducirán a buen puerto.

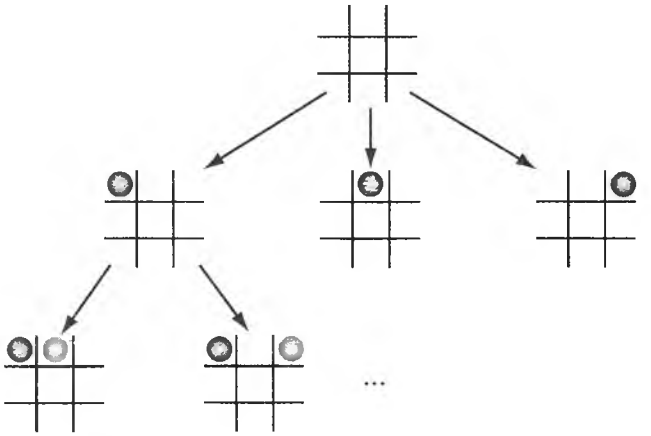
La gran diferencia entre el algoritmo de *back-tracking* y el de *branch-and-bound* es que la primera técnica recorre el árbol de planificaciones en profundidad, mientras que la segunda lo recorre en anchura, y esta diferencia es fundamental, ya que según como esté planteada la representación del problema, una poda u otra pueden tener eficiencias bastante distintas.

Ir podando el árbol a medida que se va recorriendo se hace completamente necesario, ya que, como en casi cualquier otro problema combinatorio, si no se hiciera así, el número de planificaciones y, por tanto, de ramas del árbol se haría tan enorme que al final sería imposible recorrerlo en un tiempo razonable. Para acelerar la poda, estas técnicas basadas en el recorrido de árboles suelen usar unas herramientas llamadas «heurísticas», que básicamente son la implementación de ciertas nociones intuitivas que un experto en la materia puede tener para saber cuándo una determinada rama no va a llevar a un plan viable y podarla lo antes posible. Por

EL TEOREMA DEL *NO-FREE LUNCH*

El teorema conocido como del *no-free lunch* («almuerzo no gratuito») afirma que no existe ningún algoritmo capaz de resolver todos los posibles problemas de la mejor manera posible. El problema se plantea a partir de una metáfora sobre el coste de los platos en distintos restaurantes, y de ahí su curioso nombre. Planteémonos la existencia de un determinado número de restaurantes (cada uno de los cuales representa un algoritmo de predicción determinado), con un menú que asocia cada plato (cada plato es un problema de predicción determinado) con un precio (que representa la calidad de la solución que ofrece ese algoritmo aplicado a ese problema). En este caso, tendríamos que una persona a quien le gusta comer de todo y además es algo tacaña puede investigar qué restaurante, en un momento dado, ofrece el plato que más le apetece a un mejor precio. En cambio, una persona vegetariana que acompañe a esa persona omnívora seguramente se encontrará con que el plato vegetariano es mucho más caro. La persona omnívora, si quiere un bistec, puede escoger el restaurante que lo sirva a mejor precio. Pero ¿qué pasa con la persona vegetariana que la acompaña? Resulta que el único plato vegetariano del restaurante está por

supuesto, el hecho de podar antes o después una rama inviable puede significar muchos minutos u horas de cálculo desperdiciadas, ya que el número de opciones que se deberán analizar crece de manera exponencial a medida que se va subiendo de nivel en una rama.



Ejemplo sencillo de árbol de planificación aplicado al juego del tres en raya.

las nubes, pero no tiene más remedio que elegirlo. El sufrido compañero vegetariano representa muy apropiadamente el caso en el que, dado un problema concreto, el estar obligado a utilizar un algoritmo de resolución determinado hace que los resultados sean, seguramente, mucho peores. Las reflexiones introducidas por el teorema del *no-free lunch* afectan a la planificación, ya que a pesar de los enormes esfuerzos que existen en la comunidad investigadora para diseñar un «superalgoritmo» o una «supertécnica» capaz de realizar en todo momento la mejor planificación posible, siempre acaba apareciendo un determinado conjunto de datos o un contexto para los cuales otro algoritmo o técnica resultan mejores.

Además, un corolario de este problema apunta a que si se invierten muchos esfuerzos en ajustar el rendimiento de un algoritmo para que muestre un comportamiento excelente ante unos datos, dicho ajuste está estropeando el comportamiento del algoritmo ante otro conjunto de datos diferente, con lo cual se llega a la conclusión básica de que, o se es bueno en unas pocas cosas y malo en muchas, o mediocre en todas.

Detección de conflictos

Isla de Chipre, Mediterráneo oriental, enero de 1997. Los gobiernos de Chipre y de Grecia anuncian la instalación de dos baterías de misiles tierra-aire S-300 de procedencia rusa, lo cual provoca un notable refuerzo de las fuerzas armadas chipriotas y griegas en el contexto del Espacio de Defensa Integrado.

La agencia de defensa rusa confirma públicamente que son sistemas puramente defensivos, lo cual no afectará al precario balance de poder militar en la isla entre las fuerzas turcas y las griegas.

Misiles S-300 mostrados en un desfile militar ruso (fuente: Archlinux).

Turquía, enero de 1997. De manera inmediata, el gobierno turco declara que esto es una grave amenaza para la soberanía de su país y empieza a aplicar costosísimas contramedidas para esta amenaza. Además, declara que si los misiles llegan a tocar suelo chipriota los atacará e, incluso, empezará una guerra en la isla.

Ante estas amenazas, el gobierno chipriota pone en máxima alerta a su ejército, y esta situación se prolonga hasta junio del mismo año.

Primavera de 1997. A su vez, el gobierno griego cree que la instalación de los misiles S-300 no es suficiente para contener la amenaza turca, ya que las baterías están expuestas a un golpe turco y no sobrevivirían en caso de ataque. Por tanto, Grecia considera que cualquier intento de desestabilizar la región vendrá del lado turco, ya que los misiles S-300 son puramente defensivos.

Durante este periodo, además, el ejército griego se moviliza para dar soporte al ejército chipriota en caso de ataque turco. Y Rusia se mantiene neutral pero advierte que la venta de los dos sistemas S-300 se ejecutará sin ninguna interferencia ajena.

Turquía inicia unas intensas acciones diplomáticas entre sus aliados estratégicos de la OTAN, pero éstas no fructifican, por lo que, seguidamente, decide entablar relaciones con Israel para recibir entrenamiento con el manejo de sistemas S-300.

La inestable región del Mediterráneo oriental.

Septiembre de 1997. La flota naval turca empieza a rastrear el Mediterráneo oriental en busca de buques, en especial navíos rusos, con el fin de interceptar el envío de los misiles. Por ese motivo, Rusia y Grecia advierten a Turquía de que entrarán en guerra si Chipre es atacado o bloqueado marítimamente.

Diciembre de 1997. Rusia moviliza grandes fuerzas navales en la zona, incluyendo portaaviones, submarinos, etc. Se supone que el objetivo de esta flota es transportar los sistemas S-300 y destruir la flota turca ante cualquier intento de interceptar el envío.

Enero de 1998. Ante las presiones internacionales de Estados Unidos y del Reino Unido, y frente a la amenaza de guerra turca, Grecia decide finalmente no instalar los misiles en Chipre y, a cambio, los coloca en la isla griega de Creta. En Chipre se acaban instalando otras baterías y armamento menos potente, que Grecia cede al gobierno chipriota a cambio de los S-300.

A raíz de este conflicto, que pudo haber tenido un trágico desenlace con graves consecuencias internacionales, los gobiernos turco y chipriota sufrieron una gran desestabilización. Sin embargo, no ocurrió lo mismo en el caso del gobierno griego.

Pero ¿qué pinta en todo esto la inteligencia artificial? ¿Cómo puede prevenir, predecir o incluso advertir de los movimientos tácticos más sensatos para evitar si-

tuaciones de riesgo bélico como la que se ha descrito? Pues bien, un grupo de investigadores de la Universidad de Chipre publicaron en el año 2005 un complejo sistema inteligente basado en mapas cognitivos difusos entrenados por algoritmos evolutivos capaces de predecir y simular con gran precisión situaciones de inestabilidad política. Dicho sistema, si se adapta al conflicto chipriota, recoge 16 variables que van desde la «inestabilidad/intensidad en Chipre» hasta las «influencias internacionales», pasando lógicamente por variables como el «apoyo político griego» o «el refuerzo del ejército turco».

Un mapa cognitivo difuso no es más que una red neuronal en la que cada neurona contempla la intensidad de una variable a lo largo del tiempo. Por ejemplo, en un determinado momento, el apoyo político griego puede ser bastante grande y la neurona que recoge esta variable puede contener un valor del 92%, mientras que la que contempla la variable del refuerzo del ejército turco puede ser baja en un momento determinado y estar fijada, por ejemplo, en un 23%.

Por otro lado, cada neurona está conectada con sus vecinas mediante una arista que pondera la relación causa-efecto entre las dos neuronas conectadas. Por ejemplo, la «inestabilidad política de Chipre» afecta en 0,32 al «refuerzo del ejército turco», de tal manera que si la inestabilidad política en un determinado momento es del 50%, esto provocaría un incremento directo del refuerzo del ejército del 16%, es decir, $0{,}32 \cdot 50\%$. Además, hay relaciones causales negativas como, por ejemplo, que la variable «solución al problema Chipriota» afecta en un –0,21 a la variable «inestabilidad política de Chipre».

Las complejas relaciones causales entre las neuronas (en total, en dicho mapa cognitivo existen 45 conexiones) son fijadas por un algoritmo evolutivo en el que cada uno de los individuos de la población representa una matriz de pesos que ponderan las 45 conexiones entre las variables conceptuales del mapa, y la bondad de cada matriz se mide en función de cómo es capaz de describir situaciones de escalada de tensión vividas en el pasado.

De esta manera, cuando los investigadores tuvieron entrenado el sistema con la matriz de pesos adecuada, pudieron hacer simulaciones del tipo «que pasaría si…», con el fin de buscar la mejor solución al problema chipriota. Finalmente, en un artículo publicado por los mismos autores se analizaban tres escenarios para ver qué predicciones realizaba el sistema. En uno de los casos le planteaban qué pasaría si el ejército turco abandonaba para siempre la isla, y la predicción fue toda una sucesión de situaciones de inestabilidad creciente que acababan desembocando en un escenario de caos y anarquismo.

LA INTELIGENCIA ARTIFICIAL Y LAS GUERRAS

A lo largo de la historia de la humanidad, las guerras siempre han sido un factor clave para el avance tecnológico, desde el desarrollo de la ingeniería por parte de Arquímedes durante las guerras púnicas, hasta el desarrollo de la energía nuclear en la Guerra Fría. En este sentido, la inteligencia artificial también se originó en circunstancias bélicas. Los propios fundamentos de la inteligencia artificial y de la computación se desarrollaron al calor de la Segunda Guerra Mundial, dada la necesidad de descifrar con rapidez los códigos secretos nazis. Y durante la Guerra Fría los grandes avances en este campo vinieron por la necesidad de traducir grandes volúmenes de textos técnicos y científicos del ruso al inglés. Así, para poder realizar estas traducciones de manera automática, se propició el desarrollo del procesamiento del lenguaje natural. Sin embargo, en el año 1966 se publicó el informe ALPAC (*Automatic Language Processing Advisory Committee*), en el que un comité gubernamental desaconsejó rotundamente continuar la inversión estatal en el procesamiento del lenguaje natural, dados los pobres resultados obtenidos en diez años intensivos de investigaciones.

Máquina Enigma, usada durante la Segunda Guerra Mundial para el cifrado y descifrado de mensajes.

Aunque tal vez entraría más en la categoría de «rumor» por su dudosa credibilidad, se cuenta que en la Guerra del Golfo Pérsico el ejército estadounidense bombardeó sistemáticamente todos aquellos objetivos que, según un sistema inteligente, se predecía que escondían aeronaves. Parece ser que las decisiones se tomaban de manera automática mediante unas herramientas que se detallarán más adelante, conocidas como «redes neuronales de Kohonen».

Capítulo 5

Análisis de datos

Parece ser que una importante empresa de distribución americana, Osco, encargó a su departamento de informática hace unos años que diseñara un sistema capaz de analizar los ingentes volúmenes de datos que creaba diariamente, con el fin de sacar algunas conclusiones o analizar tendencias de comportamiento del mercado.

Una vez construido el sistema, una de las primeras y más sorprendentes tendencias que descubrieron es que entre las 17:00 h y las 19:00 h las ventas combinadas de pañales y de cerveza se incrementaban de manera notable, o, en otras palabras, que de manera significativa muchos clientes que adquirían pañales en esa franja horaria también incluían cerveza en su misma cesta de la compra. Esta tendencia, en primera instancia desconcertante, se explica cuando se tiene en cuenta que los clientes con niños pequeños no pueden salir de casa por las tardes para ver los partidos de béisbol, baloncesto o fútbol, ya que tienen que cuidar a sus vástagos, de ahí que adquieran la cerveza para consumirla mientras disfrutan del partido.

Pero ¿cómo se aprovechó Osco de este descubrimiento? Pues bien, cuando se detectó dicha tendencia rápidamente se acercaron las estanterías de pañales y las de cerveza, y se dispararon las ventas combinadas de ambos productos. Cundió el ejemplo y hoy en día todas las cadenas de distribución que se precien usan herramientas de minería de datos del tipo *data warehouse* para analizar las tendencias y lanzar las ofertas de sus productos.

Toda esta jerga de «minería de datos» y «*data warehouse*» se estudiará a continuación, pero digamos que el análisis de datos es la especialidad estadística capaz de generar información a partir de datos recogidos sistemáticamente. No obstante, debido a la complejidad creciente de los datos que nos proporciona el entorno, cada vez es más complicado ejercer este análisis, hasta el punto de que hoy en día se considera una disciplina a caballo entre la estadística y la inteligencia artificial.

Se denomina «minería de datos» a la especialidad que extrae conocimiento de la información generada a partir de los datos. Aunque el análisis de datos nació en el siglo XVIII a la vez que surgían los primeros estados modernos capaces de recabar información de manera sistemática de sus sociedades y economías, la minería de datos, en cambio, no surgió hasta finales del siglo XX, cuando la potencia computa-

cional y las nuevas herramientas de la inteligencia artificial pudieron ser aplicadas a cantidades ingentes de datos para generar información y, a partir de ésta, extraer conocimiento.

La minería de datos

Un proceso típico de minería de datos da como resultado un modelo matemático que ayuda a explicar la información y las tendencias observadas a partir de los datos, pero que también puede predecir la aparición de nuevas tendencias o incluso clasificar o segmentar los datos a partir de patrones de comportamiento de identificación no trivial.

En la minería de datos se introducen datos, se procesan para generar información y de aquí se extrae conocimiento.

Las primeras herramientas de análisis de datos y, por tanto, las más fundamentales, provienen de los conceptos propuestos en el siglo XVIII por el reverendo Thomas Bayes sobre la probabilidad condicionada. De hecho, el problema que dificulta en gran medida los proyectos de análisis de datos reside en el origen mismo de los datos. Por ejemplo, imaginemos que queremos analizar datos provenientes de pacien-

tes que sufren cáncer y para ello cogemos la información del archivo de un hospital especializado en oncología. Lo normal es que tengamos disponible mucha más información de pacientes enfermos que de pacientes sanos, ya que precisamente los datos los estamos obteniendo de un lugar al que acuden personas enfermas de cáncer y no las que no lo tienen. Esta desviación inicial es lo que Bayes planteó mediante la introducción del concepto de probabilidad condicionada, ya expuesto en un capítulo anterior del presente libro. Los estudios sobre probabilidad condicionada de Bayes generaron un conjunto de herramientas que tienen en cuenta esa desviación inicial para poder compensarla y sacar conclusiones no sesgadas. En general, un proceso de minería de datos está compuesto por los siguientes pasos:

1. Selección del conjunto de datos. En este paso se seleccionan las variables que queremos segmentar, clasificar o predecir (también llamadas «variables objetivo») y las variables independientes, que son aquellos datos a partir de los cuales se construirán los modelos. Además, a menudo es imposible trabajar con todos los datos disponibles; por eso también en este paso deben seleccionarse las muestras con las que se trabajará a continuación.
2. Análisis de las propiedades de los datos. En este estadio se realiza un primer estudio simple de los datos con el objetivo de identificar valores atípicos o marginales que se salen del rango de valores razonables. También en este paso se desestiman aquellas variables que no aportan información significativa para resolver el problema en cuestión.
3. Transformación de los datos de entrada. En esta etapa los datos se suelen normalizar, ya que trabajar con datos no normalizados suele provocar errores importantes en las etapas posteriores del modelado. Por ejemplo, si dos de las variables de un problema son la estatura y el peso de las personas de un determinado país, la primera seguramente estará medida en milímetros, mientras que la segunda lo estará en kilogramos. Si a continuación se usara una red neuronal para el modelado de estos datos, las diferencias tan significativas en las magnitudes de los valores de entrada (una persona suele medir miles de milímetros, pero acostumbra a pesar decenas de kilogramos) provocarían un mal funcionamiento de las herramientas de modelado. Por ello, los datos se suelen normalizar entre 0 y 1.
4. Modelado. Ésta es la fase central de la minería de datos. De hecho, las técnicas de minería de datos se clasifican en función de la técnica o metodología usada en este paso. Por ello, bajo esta denominación se suelen englobar un gran

número de técnicas y metodologías normalmente provenientes del *soft computing* (técnicas informáticas dedicadas a solucionar problemas que manejan datos incompleto o inexactos) encaminadas siempre a la extracción de «información no trivial» o conocimiento, como las redes neuronales, SVM, etc.

5. Extracción del conocimiento. A menudo la herramienta usada en la etapa anterior no extrae el conocimiento de modo inmediato; por ello en esta etapa se aplican diversas herramientas para, por ejemplo, a partir de una red neuronal correctamente entrenada, extraer el nuevo conocimiento generado.
6. Interpretación y evaluación de los datos. A pesar del uso intensivo de herramientas computerizadas en la minería de datos, esta área de la ingeniería toda-

¿ES EL PAPA UN *ALIEN*?

En 1996, Hans-Peter Beck-Bornholdt y Hans-Hermann Dubben se preguntaban en la prestigiosa revista *Nature* si el Papa era un ser humano. Su razonamiento consistía en que, si escogemos uno de entre todos los seres humanos, la probabilidad de que éste sea el Papa es de 1 entre 6.000 millones. Y siguiendo la analogía del razonamiento silogístico, el Papa tiene 1 entre 6.000 millones de posibilidades de ser un ser humano.

La contestación a esta falacia la dieron Sean R. Eddy y David J.C. MacKay en la misma revista mediante el uso de la probabilidad bayesiana. La contestación constataba que la probabilidad de que un individuo sea el Papa sabiendo que es un ser humano no tiene por qué ser la misma que la probabilidad de que un individuo sea humano sabiendo que es el Papa. Usando notación matemática:

$$P(humano \mid papa) \neq P(papa \mid humano).$$

Si queremos conocer el valor de $P(humano \mid papa)$, debemos aplicar el teorema de Bayes, y entonces obtenemos la siguiente expresión:

$$P(humano \mid papa) = \frac{P(papa \mid humano) \cdot P(humano)}{P(papa \mid humano) \cdot P(humano) + P(papa \mid alien) \cdot P(alien)}$$

Si asumimos que la probabilidad de que un individuo (del planeta Tierra, se entiende) sea un alienígena es negligible ($P(alien) \approx 0$), la de que ese individuo sea humano tiende a 1 ($P(humano) \approx 1$), y la de que un *alien* sea escogido como Santo Padre es más bien baja ($P(papa \mid alien) < 0{,}001$); entonces, con toda probabilidad, el Papa es un ser humano ($P(humano \mid papa \approx 1$).

vía dista en gran medida de ser un proceso industrial completamente automatizable. De hecho, siempre se dice que aún se trata de un proceso muy artesanal y dependiente de la experiencia del ingeniero que lo ejecuta. Por este motivo, una vez finalizado el proceso de extracción de conocimiento, siempre se debe validar que las conclusiones obtenidas son correctas y que no nos están conduciendo o bien a un conocimiento trivial (por ejemplo, que todos los seres humanos miden entre 1,4 m y 2,4 m) o bien a un conocimiento falaz. Además, en un caso real de minería de datos, se ejecutan diversas metodologías sobre los mismos datos. En esta etapa es donde se comparan los resultados obtenidos por las diversas alternativas de análisis y extracción de conocimiento.

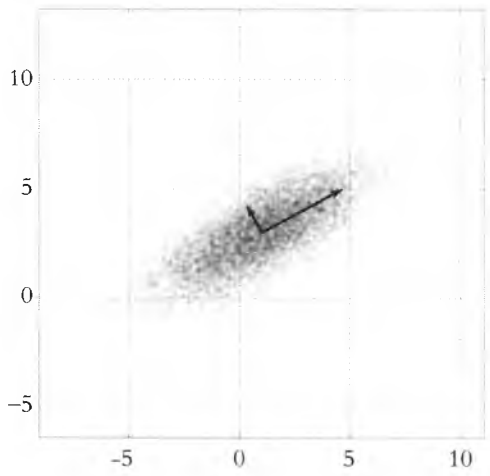
La maldición de la dimensionalidad

Es bien sabido que la intuición poco reflexionada no es buena amiga de la estadística y de la probabilidad. Uno siempre tiende a pensar que ante un problema de análisis de datos, cuantos más datos (que no más muestras) de entrada se tengan, más información y, consecuentemente, más conocimiento se podrá extraer. Sin embargo, nada más lejos de la realidad. Tanto es así que esta falacia, al ser una trampa común en la que los «mineros» principiantes suelen caer, incluso ha sido bautizada por los expertos como «la maldición de la dimensionalidad», también conocida como «el efecto de Hughes».

El problema surge al incrementarse exponencialmente el volumen de un espacio matemático al añadirse dimensiones adicionales. Por ejemplo, 100 puntos (10^2) son suficientes para muestrear un intervalo de una unidad siempre que los puntos no disten más de 0,01 entre sí. Sin embargo, en un cubo de lado la unidad, ya harían falta 1.000.000 puntos, o 10^6, y en el hipercubo de dimensión 10 y lado la unidad, 10^{20}. Por tanto, al ir añadiendo nuevas dimensiones, para poder mantener una equivalencia entre la proporción de muestras recogidas y el número de dimensiones (o en otras palabras, la densidad del espacio matemático en el que vamos a trabajar), las muestras que se tienen que analizar deben ir creciendo exponencialmente. Pongamos un ejemplo práctico. Si se quiere hacer un ejercicio de búsqueda de patrones de voto al parlamento de un país y se dispone de muchos datos sobre los votantes y sobre sus preferencias de voto, puede que algunos de esos datos no sean relevantes en la decisión del voto, como, por ejemplo, su estatura. En este caso, es mejor eliminar la variable «estatura» con el fin de que las muestras recogidas de los votantes tengan una mayor densidad dentro del volumen matemático en el que nos movemos.

Precisamente debido a la maldición de la dimensionalidad ha surgido una rama de la estadística denominada «selección de variables» (también conocida por su nombre en inglés, *feature selection)*, que usa y combina diversas herramientas matemáticas con el objetivo de eliminar el máximo número de datos que no aportan ninguna información nueva a un determinado problema. Esto puede significar desde la supresión de información redundante o correlacionada hasta la eliminación de información aleatoria y variables constantes. Por variable constante se entiende un valor que prácticamente no varía para el conjunto de las muestras. Un ejemplo sería la variable «nacionalidad» en el análisis de las tendencias de voto en la población de un mismo país. Lógicamente, esta variable será igual para todos o casi todos los votantes de ese país y, por tanto, no aporta ningún valor.

La técnica de selección de variables más utilizada es el análisis de componentes principales (ACP), que busca la proyección según la cual los datos presentan una mayor variabilidad. En el caso de la siguiente figura, las dos flechas dibujadas representan las dos componentes principales de máxima variabilidad de la nube de muestras, especialmente la flecha más larga. Por tanto, si quisiéramos reducir la dimensionalidad de los datos, podríamos sustituir las dos variables representadas en los ejes de ordenadas y abscisas por una nueva variable, que sería la proyección de las muestras sobre la componente especificada por la flecha más larga.



En esta gráfica las flechas señalan las direcciones en las que los datos muestran una mayor variabilidad (fuente: BenFrantzDale).

¿Y ÉSE QUIÉN ES? EL RECONOCIMIENTO DE CARAS

Un gran número de cámaras fotográficas actuales detectan las caras dentro de la imagen en el momento del encuadre. Hoy día existen muchos tipos de dispositivos y programas que utilizan algoritmos de detección de caras para su funcionamiento. Un ejemplo de ello son las cámaras fotográficas digitales que incluyen una función que permite detectar el número de caras en una fotografía y calibran automáticamente los parámetros de la cámara para intentar que todos los rostros queden bien enfocados. La red social Facebook también incorpora una función de detección de caras capaz de sugerir la presencia de determinadas personas de interés en las fotos que sube un usuario. ¿Cómo se logra todo ello?

La mayoría de métodos de detección de caras centran su funcionamiento en el análisis de componentes principales; se basan en el entrenamiento del sistema con conjuntos de imágenes de distintas caras, de modo que el sistema extrae los componentes principales, tanto de la cara de una misma persona como del conjunto de todas las caras. En realidad, el sistema lo que hace es memorizar los rasgos más característicos de la cara de cada persona para poder reconocerla en el futuro. De este modo, dada una nueva imagen, y aplicando de nuevo un estudio de componentes principales, el sistema compara la información extraída de esa imagen con los componentes de su conjunto de entrenamiento. Basándose en el porcentaje de coincidencia, es capaz de detectar si aquello que se está analizando es una cara o un zapato, e incluso reconocer a qué persona concreta pertenece la cara en cuestión.

El ACP busca la transformación lineal que genera un nuevo sistema de coordenadas para el conjunto inicial de muestras, en el cual la primera componente principal captura la variabilidad mayor, la segunda representa la segunda mayor variabilidad, y así hasta tantas componentes como se desee. Una de las ventajas del uso del

ACP es que en uno de los pasos intermedios en la búsqueda de las componentes de máximo crecimiento de la variabilidad se puede obtener la variabilidad explicada por cada nueva componente principal. Por ejemplo, la primera componente principal puede explicar el 75% de la variabilidad; la segunda, el 10%; la tercera, el 1%, etc. Y de esta manera se puede reducir la dimensionalidad asegurando que las nuevas dimensiones que sustituyen las características originales explican un mínimo de variabilidad de los datos (normalmente se aconseja que la variabilidad explicada por las componentes seleccionadas sume en torno al 80%).

A pesar de las bondades del ACP y de su relativa sencillez de ejecución (hoy en día todos los paquetes estadísticos ya implementan el ACP de serie), esta técnica tiene como contrapartida que implica cálculos que van creciendo en complejidad a medida que aumenta el número de dimensiones del modelo, de modo que el coste computacional puede llegar a ser inabordable. En estos casos se suele recurrir a otros dos métodos de selección de variables: el *greedy forward selection* (literalmente, «selección voraz hacia delante») y el *greedy backward elimination* («eliminación voraz hacia atrás»). Ambos presentan dos grandes desventajas: el gran coste computacional que implican y la poca seguridad que aportan de haber escogido las variables más adecuadas. Sin embargo, su fácil implementación, la sencillez del concepto en el que se basan y el hecho de que el coste computacional no sea tan alto cuando se dispone de un gran número de dimensiones como con el ACP, han popularizado su uso entre la comunidad de «mineros».

Aunque, como bien indica su nombre, uno va «hacia delante» y el otro, «hacia atrás», tanto la selección voraz hacia delante como la eliminación voraz hacia atrás

ALGORITMOS VORACES

Los algoritmos voraces son un tipo de algoritmos que siguen una filosofía muy concreta. La idea se basa en que para decidir el siguiente paso (ya sea dentro de un problema de planificación, de búsqueda o de aprendizaje) siempre se escoge la opción que maximiza, a corto plazo, un determinado gradiente, sea cual sea el problema que se ha de resolver. La bondad de los algoritmos voraces es que son capaces de maximizar muy rápidamente una determinada función matemática. Pero, en contrapartida, en funciones complejas y multimodales (es decir, con varios máximos) suelen acabar estancándose en un máximo local, ya que no poseen la capacidad de tener una visión global del problema. Esto acaba siendo una mala estrategia, ya que la optimización se suele quedar en un subóptimo.

usan la misma filosofía. Pero la mejor manera de explicar ambos métodos es a través de un ejemplo práctico. Imaginemos que queremos seleccionar las variables que mejor explican las tendencias de voto al parlamento de un país. Entre las cinco variables conocidas de la muestra recogida tenemos: poder adquisitivo, ciudad de origen, formación académica, sexo y estatura del votante. Y la herramienta que usaremos para el análisis de las tendencias es una red neuronal. Para empezar, utilizando el método de selección voraz hacia delante, se escoge la primera de las variables del problema y, sólo con esa variable, se modelan los datos mediante la red neuronal. Una vez construido el modelo, se evalúa su tasa de predicción y se guarda la información. El proceso se repite exactamente igual con la segunda variable, y luego con las tres variables restantes. Cuando ya se ha hecho todo el análisis, se escoge la variable cuyo modelo asociado presentaba mejores resultados y se repite el proceso de modelado mediante la red neuronal y la evaluación del modelo, pero esta vez con dos variables. Suponiendo que la variable que hubiera dado mejores resultados hubiera sido la formación académica, se probarían todos los conjuntos de dos variables en los que la primera fuera formación académica. Así, tendríamos el modelo «formación académica y ciudad de origen», el modelo «formación académica y sexo» y el modelo «formación académica y estatura». De nuevo, una vez analizadas las cuatro combinaciones, se selecciona la mejor, por ejemplo, «formación académica y poder adquisitivo», y se vuelve a repetir el proceso con tres variables, donde ahora las dos primeras están fijadas. El proceso continúa hasta que, al añadir una nueva variable, la bondad del modelo no mejora con respecto a la bondad del modelo con una variable menos.

La eliminación voraz hacia atrás funciona justo al revés, es decir, partiendo del modelo que incorpora todas las variables, va eliminando, una a una, aquellas que no empeoran la bondad del modelo.

Como se puede apreciar, a pesar de la sencillez del método, no deja de ser una estrategia «poco inteligente» porque no asegura que se encuentre la mejor combinación de variables, a la vez que supone un alto coste computacional al tener que modelar los datos en cada paso de selección o eliminación de variables.

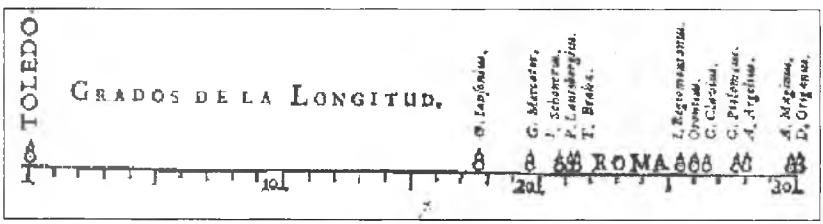
El hecho de que los métodos de selección de variables existentes tengan importantes contrapartidas provoca que constantemente se presenten nuevos métodos en los foros especializados. Estos nuevos métodos suelen seguir la filosofía del ACP, es decir, buscan nuevas variables que sustituyen a las originales y aportan una mayor «densidad» o «riqueza» en la información. Este tipo de variables son conocidas como «variables latentes». En general, son muy utilizadas por un gran número de

disciplinas, aunque, probablemente, donde mejor se aplican es en las ciencias sociales. Descriptores como la calidad de vida de una sociedad, la confianza del mercado o la capacidad espacial de una persona son variables latentes que no se pueden observar directamente, sino que se miden e infieren a partir de la medida de un cúmulo de otras variables más tangibles. Además, estas variables latentes tienen la ventaja de que aúnan distintas variables en una sola, de modo que reducen la dimensionalidad del modelo y lo hacen más manejable.

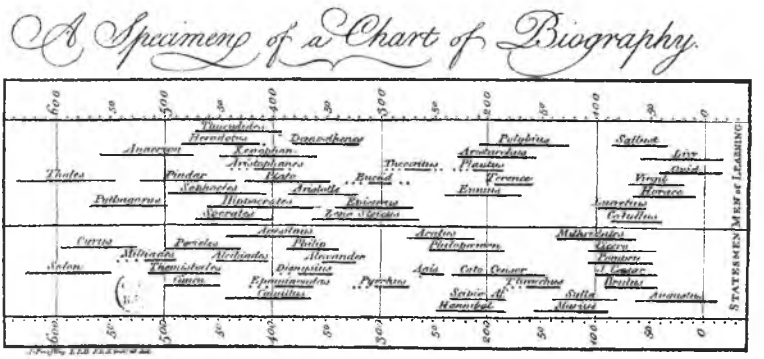
Visualización de datos

La visualización de datos es la rama de la ingeniería que estudia de qué forma los datos numéricos, por lo general multidimensionales, se pueden representar gráficamente para ser visualizados por un ser humano. Por eso, al igual que el análisis de datos, la visualización de los mismos se popularizó cuando los incipientes estados modernos pudieron generarlos de manera sistemática sobre la evolución de sus economías, sociedades y sistemas productivos. De hecho, esta rama de la ingeniería es vecina, o incluso está superpuesta, a la del análisis de datos, ya que muchas de las herramientas, metodologías y conceptos usados para facilitar su visualización surgen de su análisis, y viceversa.

Probablemente, la primera visualización de datos estadísticos de la que se tiene registro es la realizada por Michael van Langren en 1644, en la que se muestran las 12 estimaciones, realizadas por 12 científicos distintos, de la longitud entre Toledo y Roma. La palabra «ROMA» señala la propia estimación de Langren, y la pequeña flecha borrosa que aparece más o menos en la parte inferior central de la línea es la longitud correcta estimada con métodos modernos.

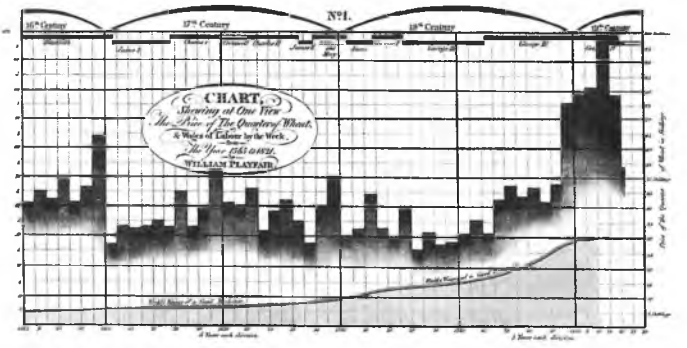


Ya en el siglo XVIII, Joseph Priestley generó el siguiente gráfico para representar en qué momento vivieron algunos de los hombres más sobresalientes de la Antigüedad:



En ese mismo siglo, gracias a las reflexiones filosóficas de Immanuel Kant en las que se afirmaba que es la representación la que hace al objeto posible y no al revés, se forjó la conciencia de que no se puede discutir sobre el conocimiento o la realidad sin tener en cuenta que es la mente humana la que construye esa realidad o conocimiento. Esto colocó la ciencia de la representación y la visualización de datos en el lugar de máxima importancia que merecía.

Más tarde, durante la Revolución Industrial, empezaron a aparecer ya otros métodos de representación de datos más sofisticados, como los introducidos por William Playfair para representar la producción industrial y económica mediante la evolución de los precios del trigo y de los salarios a través de los distintos gobiernos y a lo largo de más de 250 años:



A partir de aquí, y ya de la mano de las ciencias de la computación, los profesionales en la visualización de datos empezaron a trabajar en entender cómo tiene que ser una buena representación de los datos para que un analista la pueda interpretar

rápida y fácilmente. Uno de los aspectos más importantes que deben tenerse en cuenta (más que el propio componente técnico de los datos, el modelo de representación y el «motor gráfico» usado para visualizarlos) es la limitación perceptiva del analista, el consumidor final de los datos. Cuando éste intenta entender una representación visual de los datos ejecuta unos determinados procesos cognitivos que construyen su modelo mental de los datos. Pero estos procesos cognitivos tienen fuertes limitaciones perceptivas, como, por ejemplo, que la mayoría de mortales somos incapaces de entender mentalmente más de cuatro o cinco dimensiones, y estas limitaciones deben tenerse en cuenta para facilitar la construcción de los modelos. Por todo ello, una buena visualización de los datos tiene que mostrar información de manera jerárquica a diferentes niveles de detalle, ser coherente y evitar al máximo cualquier posible distorsión en las representaciones. Además, debe minimizarse el impacto de los datos que no aportan información o que pueden conducir a conclusiones erróneas, y añadir otros estadísticos que aporten información sobre la significancia estadística de cada parte de la información.

Para conseguir todo esto se recurre a estrategias similares a las que se han visto en el capítulo dedicado al análisis de datos. La primera es la de reducir dimensiones, lo que se consigue mediante métodos ya vistos anteriormente, como la proyección de los modelos en variables latentes. La segunda es la de reducir el número de muestras del modelo mediante su clasificación en grupos significativos, proceso que se denomina *clustering* (*cluster* puede traducirse por «racimo»).

Un análisis de *clustering* consiste en dividir un conjunto de observaciones en subconjuntos (también llamados *clusters*), de tal manera que todas las observaciones que están ubicadas en el mismo *cluster* comparten ciertas propiedades, que no tienen por qué ser obvias. Mediante la «clusterización» de los datos se puede facilitar enormemente tanto su representación gráfica como su entendimiento por parte del visualizador humano, debido a la simplificación que el *clustering* introduce en la representación. Existen muchos algoritmos de *clustering*, cada uno con distintas propiedades matemáticas que los hacen más o menos aptos para un determinado tipo de datos.

Reconocimiento de patrones

No podríamos terminar un capítulo sobre análisis de datos sin hablar del reconocimiento de patrones (*patterns* en inglés), ya que uno de los grandes objetivos del análisis de datos es precisamente poder reconocer e informar sobre patrones presentes con el fin de predecir tendencias futuras. Para el reconocimiento de patrones se

pueden usar todas las herramientas comentadas hasta el momento: redes neuronales, máquinas de soporte vectorial, análisis de componentes principales, etc., y, como se verá, es una rama del análisis de datos fuertemente ligada al aprendizaje artificial.

El objetivo de un sistema clasificador, al igual que una red neuronal o una SVM, es, dada una muestra, poder predecir su clase, o en una palabra, clasificarla. Por ello, inicialmente, se le debe proporcionar al sistema clasificador un conjunto de muestras de clase conocida con el fin de que el sistema pueda aprender. Una vez que el sistema ya está entrenado, se le pueden pasar las nuevas muestras para que las clasifique. Al igual que los métodos anteriores, el conjunto de muestras iniciales de clase conocida suele dividirse en dos subconjuntos, el de entrenamiento y el de test, que sirven para comprobar, a posteriori, que el sistema no está sobreentrenado.

Los sistemas clasificadores pueden ser de dos tipos: de Michigan (llamados así porque fueron precisamente investigadores de esta universidad quienes defendieron este modelo) o de Pittsburg (por el mismo motivo). Un sistema clasificador de Michigan no es más que un algoritmo evolutivo en el que los individuos que van evolucionando son reglas, y cada regla está formada por un conjunto de condiciones y un objetivo. Y la idea es que si una muestra coincide con las condiciones impuestas por una regla, la clase de esta muestra será la indicada por el objetivo de la regla.

En cambio, en los sistemas clasificadores de Pittsburg cada individuo es un conjunto de reglas y la bondad del individuo se evalúa a partir de la tasa de error media de cada una de esas reglas contenidas. Ambos sistemas, bastante complementarios entre sí, tienen sus ventajas e inconvenientes. En los últimos treinta años, investigadores de todas las escuelas han ido proponiendo mejoras y variantes a ambos esquemas para ir puliendo sus ineficiencias.

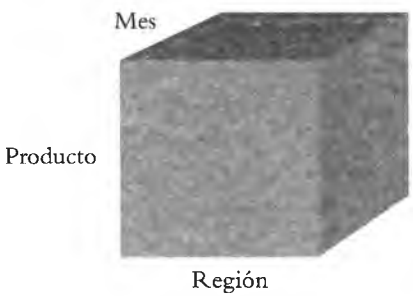
Un ejemplo práctico: análisis de ventas

Otra de las grandes aplicaciones de la inteligencia artificial aplicada a los negocios es el *datawarehousing*. Un *data warehouse* (literalmente, «almacén de datos») es una herramienta empresarial muy extendida entre los negocios que cuentan con una gran base de clientes y, por tanto, una gran base de muestras desde donde inferir tendencias, modas o patrones de comportamiento. Es el lugar donde se juntan los datos de toda la empresa, ya sean de ventas, de producción, de las campañas de marketing, de fuentes externas, etc. En la actualidad se pueden encontrar ejemplos de usos de *data warehouses* en negocios tan dispares como la banca, los hospitales, los distribuidores de alimentos, los productores de derivados del petróleo, las organizaciones gubernamentales, etc.

Crear y estructurar un *data warehouse* es una tarea compleja que puede llevar meses o años a ingenieros especializados. Sin embargo, una vez que estos almacenes de datos están construidos, estructurados y se ha asegurado su consistencia, las tecnologías de *datawarehousing* utilizan un concepto llamado «cubos OLAP» (aunque en realidad son hipercubos), que procede al estudio y al análisis de los datos. Un cubo OLAP (que responde al acrónimo de los términos anglosajones *OnLine Analytical Processing*, o «procesamiento analítico en línea») es una estructuración multidimensional de los datos que permite realizar consultas cruzadas de manera muy rápida entre datos de distinta naturaleza. Podría verse como la extensión en más dimensiones de una hoja de cálculo. Por ejemplo, si construimos una tabla en una hoja de cálculo donde representamos qué productos lácteos hemos vendido en distintos países en el año anterior, expresado en miles de unidades, podemos obtener una tabla como la que sigue:

	España	Italia	Francia	Alemania
Yogur natural	4.540	5.312	5.429	10.234
Yogur de limón	8.988	14.543	11.234	26.342
Yogur de fresa	12.349	16.234	15.345	23.387
Yogur líquido	1.676	2.221	3.234	1.476
Natillas	4.678	6.934	4.343	1.893
Arroz con leche	5.122	7.300	8.345	345
Cuajada	567	145	0	0

A continuación, nos interesaría poder desglosar estos datos en meses, con lo cual deberíamos añadir una tercera dimensión a la tabla, en la que para cada región y cada tipo de producto tengamos una división en los doce meses del año.



De esta manera, una vez que se ha estructurado el cubo, se pueden llevar a cabo complejos análisis de datos teniendo en cuenta esa estructuración precalculada. De hecho, el principal coste computacional de un *data warehouse* no es el propio análisis de los datos (en el que se suelen aplicar muchas de las herramientas comentadas a lo largo del capítulo) sino la construcción de muchísimos hipercubos con todas las dimensiones que puedan soportar los datos de la organización, teniendo en cuenta las múltiples combinaciones posibles. Por ello, la generación de los cubos OLAP es un proceso que las organizaciones suelen realizar por las noches, y que consultan y analizan el día posterior.

Con todo ello, los analistas de una organización encargada de la fabricación de productos lácteos podrían introducir en el sistema las condiciones atmosféricas de cada uno de los días del año en cada una de las regiones donde opera. Con esta nueva dimensión se pueden realizar estudios de tendencia del consumo de los diferentes productos en función de la temperatura ambiental de cada día del año en cada región.

A continuación, con este conocimiento y sabiendo las predicciones meteorológicas de otro año, los analistas pueden predecir el número de unidades que se deben producir en cada región para minimizar el almacenamiento innecesario de productos lácteos, que tiene un alto coste debido a la cadena de frío que se debe mantener a lo largo de la vida del producto.

Para complicar un poco más el concepto de las dimensiones de un cubo OLAP, a menudo, dentro de una misma dimensión se introducen jerarquías. De este modo, continuando con el caso anterior, en la dimensión temporal se puede introducir una nueva jerarquía inferior al mes, que puede ser el día, y una superior, como, por ejemplo, el trimestre, porque seguramente no se consumen el mismo número de unidades lácteas en invierno que en verano, ni a principio que a final de mes. Otra dimensión que se podría jerarquizar es la región, y se podría introducir una jerarquía superior que abarcara, por ejemplo, el sur de Europa, la Europa central, etc., o bien una inferior, como puede ser Lombardía, Bretaña, Andalucía, etc.

Por supuesto, una vez construidos los cubos OLAP, aparte del obvio análisis de datos, del que ya hemos hablado, se pueden realizar otras múltiples operaciones de visualización de los mismos. Por ejemplo, se podrían visualizar «rebanadas» bidimensionales del cubo, visualizar «cubitos», es decir, pequeñas porciones multidimensionales del cubo, sumar o restar información a través de las jerarquías, o incluso rotar el cubo para poder observar los datos desde otro punto de vista.

MICROSOFT RESEARCH

En la actualidad, el mayor centro privado y no académico del mundo que investiga sobre inteligencia artificial es el Microsoft Research. Allí se encuentran prestigiosos científicos de renombre internacional que investigan temas tan importantes para esta disciplina como el aprendizaje artificial o nuevas interacciones hombre-máquina. Microsoft Research dispone de delegaciones por prácticamente por todo el mundo, como, por ejemplo, en Alemania, Estados Unidos, Reino Unido, China, India o Egipto.

Un área en la que este centro es líder de investigación mundial es, más concretamente, en el uso de redes bayesianas y de otras herramientas probabilísticas para temas tan importantes como la detección de correos no deseados (el conocido como *spam*) o la adaptación inteligente de las interfaces de los sistemas operativos a los patrones de comportamiento de los usuarios, de tal manera que las interfaces de usuario del futuro puedan adaptase inteligentemente a la manera de trabajar de cada persona.

Capítulo 6

Vida artificial

Vida e inteligencia son dos conceptos igualmente difíciles de definir, no sólo en el ámbito de la biología sino también desde el punto de vista filosófico. Probablemente, buscar formalismos para definir la vida es tan complejo como la búsqueda de definiciones formales para la inteligencia. Acuérdese el lector de todas las disquisiciones filosófico-matemáticas del primer capítulo para delimitar el concepto de inteligencia: el test de Turing, la habitación China, las discusiones sobre la creatividad, etc. Sin embargo, uno de los autores más reputados y activos en este campo, John H. Holland (n. 1929), que también fue el artífice de los algoritmos evolutivos, ha profundizado a lo largo de los años en la cuestión, llegando a conclusiones que nos ayudarán a comprender dicho concepto.

La vida artificial está estrechamente unida a otro importante concepto dentro de la inteligencia artificial, el *soft computing* (literalmente, computación blanda, aunque siempre se utiliza el término en inglés). El *soft computing* es un conjunto de herramientas, por lo general inspiradas en procesos presentes en la naturaleza, que resuelven problemas de gran complejidad, ya que la información que manejan es aproximada, incierta e incompleta. Algunas de las herramientas que hemos visto a lo largo de este libro, como los algoritmos evolutivos, las redes neuronales, la lógica difusa, etc., forman parte de él.

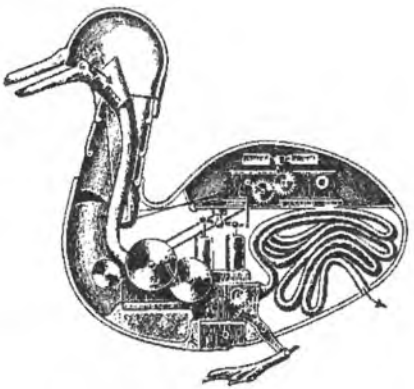
El *soft computing* se convirtió en una rama formal de la informática en la década de 1990, y en la actualidad se usa en la resolución de problemas para los cuales los expertos se han resignado a no encontrar la mejor solución, que en algunos casos podría suponer años de cálculos o tener en cuenta información imposible de recopilar, en favor de la posibilidad de encontrar una buena solución de manera rápida. Todas las áreas de la ciencia y la ingeniería modernas, desde la biología hasta las ciencias políticas, usan hoy día el *soft computing* para solucionar problemas.

Introducción a la vida artificial

Uno de los conceptos más importantes que deben existir en un sistema donde haya «vida» es que se den ciertas condiciones de emergencia para que sistemas autoorga-

nizados y mucho más complejos que las partes puedan emerger del medio. Un buen ejemplo de ello son las colonias de hormigas, donde del comportamiento relativamente simple de unos sencillos elementos, las hormigas, emerge un sistema completamente autoorganizado, la colonia, que, por supuesto, es mucho más complejo que la suma de las partes.

Otra de las características de la vida consiste en que la entidad supuestamente «viva» también debe ser capaz de sobrevivir a las condiciones del medio y, en el mejor de los casos, poder llegar a reproducirse. Asimismo, para poder decir que algo está vivo tiene que apreciarse un determinado dinamismo no aleatorio e independiente del posible cambio de leyes que gobiernan el medio. Además, un sistema vivo debe presentar un comportamiento emergente y recurrente, sin entrar, claro está, en la regularidad. Es decir, si hay un proceso que ha emergido, presenta un comportamiento apreciable, pero es un comportamiento cíclico o de bucle, la entidad que lo manifiesta no podría considerarse como «viva».



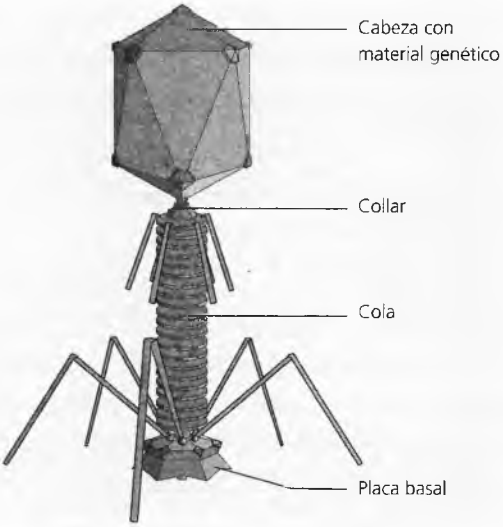
Pato de Jacques Vaucason (1709-1782), probablemente el primer ingeniero interesado en la vida artificial.

Como el lector puede apreciar, no es fácil definir con cierta precisión matemática la vida, al igual que pasaba con la inteligencia, y de aquí que no haya aún un criterio sencillo para determinar si una entidad artificial o natural puede ser considerada viva. Este debate, como se comentaba anteriormente, va más allá de las matemáticas y la computación. De hecho, en biología aún no hay pleno consenso entre los científicos sobre algunas cuestiones relativas a la vida, como, por ejemplo,

VIRUS Y PRIONES

Los virus son sistemas biológicos que por sí solos no son capaces de reproducirse; por ello, la gran mayoría de biólogos opinan que no son seres vivos, aunque no hay consenso sobre ello. Los virus pueden ser de muchos tipos diferentes, pero todos tienen en común la presencia de material genético, que inyectan en un huésped tras introducirse en él, consiguiendo que éste replique copias del virus, incluyendo, por supuesto, dicho material genético. Estas copias se van esparciendo por el organismo para infectar otras células huésped.

Por su parte, un prión es una entidad aún más simple, sin material genético, pero que tiene la propiedad de propagarse entre los organismos. Los mecanismos de trasmisión de los priones aún no han sido clarificados por la comunidad científica, pero en estos momentos son unas entidades de alto interés científico, ya que son los responsables de graves enfermedades, como, por ejemplo, la encefalopatía espongiforme bovina, conocida como «mal de las vacas locas». Los priones son proteínas naturales que tienen un plegamiento anómalo. Cuando una de esas proteínas entra en contacto con otra en el organismo, provoca un mal plegamiento de la nueva proteína contactada, que deja de ser funcional, y a la vez es capaz de propagar esta «infección» de malos plegamientos a otras proteínas. ¿Es un prión una entidad viva? Según los biólogos, no.



Esquema básico de un virus.

si un virus es una entidad viva o no lo es. De hecho, si consideramos estas definiciones, ¿es un virus informático una entidad viva? Si analizamos la situación, los virus informáticos presentan un comportamiento dinámico, apreciable, no regular ni cíclico. Sin embargo, ¿podemos afirmar que un virus informático ha surgido de forma natural? Probablemente no, ya que en última instancia ha sido un malvado programador informático el que lo ha creado, a diferencia de un virus natural que sí que habrá emergido naturalmente.

COMPUTACIÓN VIVA

El presente capítulo muestra sistemas artificiales que imitan el comportamiento de seres vivos, pero también trata justo lo contrario, es decir, sistemas de computación construidos a partir de entidades vivas. La unidad de proceso de un computador moderno está constituida por centenares de millones de transistores, que son las unidades que, mediante impulsos eléctricos, ejecutan todas las operaciones. Los transistores son objetos sin vida creados a partir de elementos inorgánicos, como el silicio. Pero ¿es posible sustituir los transistores, simples unidades metálicas, por sistemas vivos creados a partir de células? En los últimos tiempos, investigadores especializados en biología y física están logrando precisamente eso, que sean células vivas las que computen dichas operaciones matemáticas, como lo hacen los transistores. Por tanto, en el futuro se podrán implementar sistemas de vida artificial soportados en una base de computación biológica. ¿Serán los ordenadores del futuro seres vivos a los que debamos alimentar con comida en vez de electricidad?

Sistemas adaptables complejos

Para simplificar la definición del concepto de vida algunos expertos han ideado otro concepto más general, el de «sistema adaptable complejo». Un sistema adaptable complejo es un agente o conjunto de ellos que trabajan de manera coordinada y que son lo bastante inteligentes como para poder adaptarse al medio según el comportamiento que otros sistemas vayan desarrollando. De hecho, en la definición de sistema adaptable complejo encajan sistemas vivos que van mas allá de lo que a uno le puede venir a la cabeza cuando piensa en un «ser vivo», como, por ejemplo, el sistema inmune que hay dentro de un ser humano, una corporación mercantil o un ecosistema entero. El caso de la corporación es bastante curioso, ya que seguramente nadie hubiera dicho que una entidad legal como puede ser una empresa

tenga «vida». Pero si se piensa detenidamente, una corporación es una entidad que nace, crece, puede reproducirse y puede morir. En la mayoría de países europeos una corporación tiene casi los mismos derechos y obligaciones que una persona, hasta el punto de que ambos comparten definición: unas son personas físicas (los seres humanos), y las otras, personas jurídicas (las corporaciones).

SIMCITY Y LOS SISTEMAS ADAPTABLES COMPLEJOS

Las ciudades son buenos ejemplos de sistemas adaptables complejos, ya que presentan e implementan con creces las propiedades que los definen. De hecho, el juego de ordenador *SimCity*, mediante el cual se pueden construir y simular ciudades, es un magnífico entorno de pruebas para familiarizarse con un sistema adaptable complejo, ya que el mismo juego «rellena» la ciudad con ciudadanos, genera actividad social y mercantil dentro de ella y nos plantea complejas situaciones que hay que resolver, como colapsos de las vías de comunicaciones o desastres naturales.

Otro juego de ordenador con el cual podemos familiarizarnos con los sistemas adaptables complejos es *Civilization*, en el cual el objetivo es construir una civilización competitiva entera, con sus ciudades, redes de comunicaciones, tratados comerciales, sistemas defensivos, políticas sociales y científicas, etc.

Imagen de una ciudad construida con el juego SimCity.

Existe cierto consenso en considerar que son siete características las que definen un sistema adaptable complejo: cuatro propiedades y tres mecanismos. Adicionalmente, la combinación de estas características básicas da lugar a otras propiedades y mecanismos compuestos. La definición de un sistema adaptable complejo es algo más general que el propio sentido biológico de «vida», ya que nunca diríamos que una entidad financiera o una ciudad es un «ser vivo». Por eso, el término difuso y difícil de definir de «vida artificial», más allá de las noticias sensacionalistas y los foros no especializados, no suele utilizarse.

Primera propiedad: la agregación

La agregación es la suma del comportamiento de entidades simples, de modo que de ella emerge un comportamiento agregado bastante más complejo que la suma individual de las partes (piénsese en el ejemplo del hormiguero y las hormigas que lo componen, donde la adaptabilidad del hormiguero entero a los cambios del medio es mucho mayor que la adaptabilidad individual de las hormigas). Cada una de estas partes sencillas se denomina agente.

Además, esta propiedad es recursiva, y un agente que ha emergido como la agregación de otros agentes más simples puede volver a agregarse con otros, tanto de su propia especie como de otras, para formar otro agente agregado de segundo nivel. Por ejemplo, la agregación del comportamiento y productividad de todas las empresas de un país, más el comportamiento del consumo familiar, más el de las administraciones públicas, forma el producto interior bruto de un país.

Se debe puntualizar que un elemento no tratado en la agregación pero imprescindible para que puedan emerger comportamientos adaptables al medio es la comunicación entre los elementos de una categoría (primera acepción de agregación) o entre las distintas partes que forman la entidad de grado superior (segunda acepción).

Primer mecanismo: el etiquetado

El etiquetado es un mecanismo que facilita de manera activa la agregación de agentes. El etiquetado de agentes es un concepto tan sencillo como la fijación de marcas o signos, y no sólo facilitan su identificación, sino que ayudan a romper las simetrías que a menudo se forman en la agregación de sistemas complejos. Por ejemplo, si una bola blanca de billar empieza a rotar en una dirección y en su superficie no hay impreso ningún signo, el espectador difícilmente podrá determinar que la bola está

rotando, y mucho menos la velocidad de rotación. Si, en cambio, se imprime algún signo en el algún lugar de su superficie, excepto en los dos puntos por donde el eje de rotación intersecta la superficie de la bola, el espectador podrá discernir el sentido y la magnitud de la rotación.

Existe una gran variedad de signos de los que hacen uso los agentes agregados, desde los estandartes en forma de águila que los legionarios romanos empleaban para marcar cada una de las legiones hasta las complejas etiquetas con las que los modernos dispositivos de telecomunicaciones marcan las tramas de los mensajes trasmitidos, en los que no sólo se señala el orden de cada trama para reconstruir el mensaje cuando todas ellas llegan al receptor, sino que también pueden incluir sofisticados mecanismos para identificar posibles errores que hayan podido afectar al mensaje o a la misma etiqueta durante el proceso de transmisión. Por supuesto, no todas las etiquetas tienen que ser visibles; por ejemplo, los mamíferos de los distintos sexos de algunas especies se etiquetan en la época de celo usando unas sustancias químicas llamadas feromonas.

Las etiquetas facilitan a los agentes la interacción selectiva, mediante la cual éstos pueden discernir entre diversas instancias de una misma clase de agentes o las diversas partes agregadas de un agente. Esto, por supuesto, da pie a la implementación de filtros, especiación o esquemas cooperativos. También los agentes pueden mantenerse agregados, aunque las diversas partes agregadas que conforman el agente de nivel superior vayan cambiando, pero el etiquetado se mantenga. En definitiva, el etiquetado es un mecanismo-herramienta que facilita la organización y la comunicación entre agentes.

Segunda propiedad: la no-linealidad

Es poco conocido que la mayor parte de las herramientas que las matemáticas nos proporcionan son herramientas lineales: desde la aritmética hasta la topología algebraica, pasando por el cálculo diferencial, todo se basa en asunciones de linealidad. Una función es lineal si el valor de ésta, para cualquier valor asignado a sus argumentos, es simplemente la suma ponderada de la suma de estos valores. Por ejemplo, la función $4x+2y-z$ es lineal; en cambio, $4\,\mathrm{sen}x-2y^{-z}$ no es lineal.

El uso de instrumentos lineales es tan importante en las matemáticas y la ingeniería que, de hecho, en la actualidad gran parte de la actividad profesional de un ingeniero o de un científico se basa en buscar funciones lineales que aproximen con la mayor precisión posible los fenómenos naturales. Sin embargo, desafortunada-

mente, ninguna de estas herramientas funciona bien en los sistemas adaptables complejos. De hecho, uno de los conceptos que mejor definen estos sistemas es que su comportamiento global es bastante más complejo que la suma individual de las partes, y he aquí que aparece, por definición, la no-linealidad.

Un buen ejemplo que ilustra bien la no-linealidad de la naturaleza y de los sistemas adaptables complejos son las dinámicas productor-consumidor y, concretamente, el caso particular de presa-depredador. Imagínese un monte donde hay D depredadores (por ejemplo, zorros) y P presas (por ejemplo, conejos). Si la probabilidad de que un zorro cace un conejo es c, entonces cada día hay $c \cdot P \cdot D$ conejos cazados. Por ejemplo, si $c = 0{,}5$, $D = 3$ y $P = 10$, entonces hay $c \cdot P \cdot D = 0{,}5 \cdot 3 \cdot 10 = 15$ capturas. Sin embargo, si el número de zorros y conejos se cuadriplica, el número de capturas no lo hace: $c \cdot P \cdot D = 0{,}5 \cdot 12 \cdot 40 = 240$ y, como se ve, la actividad depredadora no puede obtenerse simplemente añadiendo los nuevos depredadores a las presas.

ECUACIONES DE LOTKA-VOLTERRA

Las ecuaciones del ejemplo de los zorros y los conejos se pueden complicar bastante más. De hecho, un investigador llamado Alfred J. Lotka describió qué pasaría con esas ecuaciones si empezamos a tener en cuenta las variaciones de los depredadores y las presas a lo largo del tiempo. Supongamos que $D(t)$ y $P(t)$ es el número de depredadores y presas que hay, respectivamente, en el instante de tiempo t. Además, podemos decir que en cada instante de tiempo pueden nacer n depredadores y morir m de éstos. Por tanto, la fórmula de la evolución de los depredadores a lo largo del tiempo es $D(t+1) = D(t) + nD(t) - mD(t)$, y lo mismo pasaría para las presas, es decir, $P(t+1) = P(t) + n'P(t) - m'P(t)$. Hay que tener en cuenta que, en el caso de los depredadores, un mayor número de presas supone más nacimientos; ello se expresa, por ejemplo, a través de la constante r, que describe la eficiencia de transformar la comida en descendencia. El número de encuentros presa-depredador, como vimos antes, es cPD; por consiguiente, la nueva ecuación de los depredadores quedaría:

$$D(t+1) = D(t) + nD(t) - mD(t) + r[cP(t)D(t)].$$

Sin embargo, con las presas pasa justamente lo contrario, ya que cada vez que hay un encuentro entre presa y depredador, el número de las primeras decrece; por tanto, su ecuación quedaría:

$$P(t+1) = P(t) + n'P(t) - m'P(t) - r[cP(t)D(t)].$$

Incluso en una situación relativamente simple, una no-linealidad puede afectar enormemente a un sistema agregado. Por eso se dice siempre que el comportamiento agregado de un sistema adaptable complejo es más complicado que el comportamiento individual de las partes que lo forman.

Tercera propiedad: la formación de flujos

Los flujos aparecen en todos los niveles dentro de los sistemas adaptables complejos, donde siempre tiene que haber nodos, transportadores y el recurso transportado. Sólo por poner dos ejemplos de sistemas adaptables complejos donde hay flujos podemos hablar del sistema nervioso central de un ser vivo, donde los nodos son las neuronas, los transportadores son las conexiones sinápticas entre ellas y el recurso transportado son los impulsos eléctricos; un segundo ejemplo serían los flujos dentro de un ecosistema, donde los nodos son las especies, el transportador, la cadena

Si ahora tomamos estas dos ecuaciones, fijamos las constantes y las vamos resolviendo instante a instante, veremos que $D(t)$ y $P(t)$ van oscilando a lo largo del tiempo, y las presas y depredadores van pasando por ciclos continuos de abundancia-hambruna.

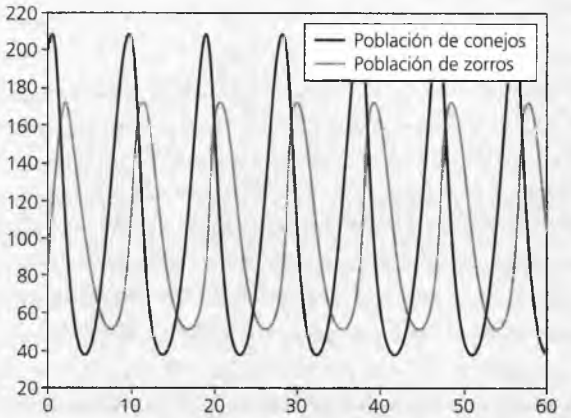


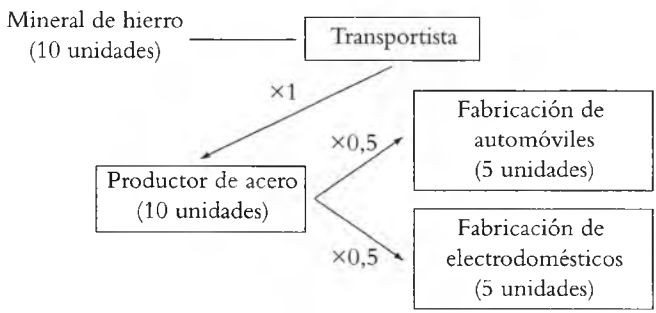
Gráfico que representa la evolución de las poblaciones de conejos y zorros a lo largo del tiempo según las ecuaciones de Lotka-Volterra.

trófica y el recurso transportado es la energía representada a través de elementos bioquímicos (proteínas consumidas, azúcares, etc.).

En general, los nodos son procesadores del recurso, y las conexiones definen las interacciones entre ellos. Sin embargo, en un sistema adaptable complejo se debe tener presente que la red de interacciones puede ser cambiante, y los nodos y conexiones pueden ir apareciendo y desapareciendo. Justamente esto es lo que hace que un sistema adaptable complejo sea una entidad adaptable al medio y que pueda, de manera autónoma, ir modificando su comportamiento según las necesidades, adecuadas o no, del momento.

El etiquetado es uno de los mecanismos de los sistemas adaptables complejos de mayor importancia para la definición de flujos; de hecho, las etiquetas pueden fijar cuáles son las conexiones críticas para el transporte de recursos.

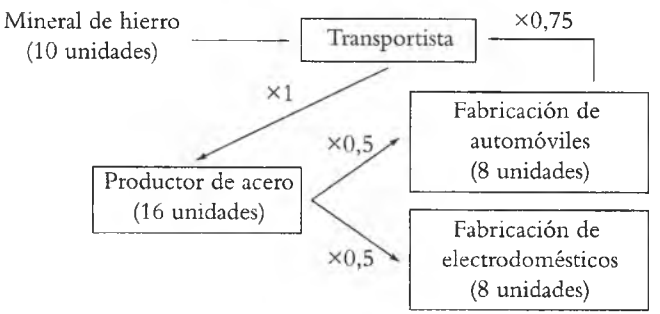
Los flujos presentan dos propiedades que son interesantes para el funcionamiento de los sistemas adaptables complejos. La primera de ellas es el efecto multiplicador que introducen en el sistema; por ejemplo, en un sistema adaptable complejo, como la economía de un país, el efecto del transporte de dinero de un nodo a otro (como el que hay entre bancos) va ejerciendo de multiplicador de riqueza. La segunda propiedad interesante es la capacidad de creación de ciclos, con el fin de que puedan darse casos de reciclaje. Por ejemplo, obsérvese en el siguiente esquema cómo aumenta de manera no-lineal la producción industrial en un sistema adaptable complejo que sería una cadena de producción de coches, con y sin reciclaje.



En un primer escenario, el productor de acero transforma, con una eficiencia del 100% (o sea × 1), el mineral de hierro en acero. A continuación, el 50% (o sea × 0,5) de la producción de acero se emplea en la fabricación de automóviles, y el otro 50%, en la de electrodomésticos. Si, para simplificar, suponemos que por cada unidad de

acero se genera un auto o un electrodoméstico, tendremos que al final del flujo se habrán producido 5 unidades de coches y 5 de electrodomésticos.

Veamos ahora un escenario con el reciclaje como agente multiplicador:



En este segundo escenario se reciclan el 75% de los coches; por tanto, el productor de acero ahora es capaz de producir más acero, que al final se convierte en más unidades de automóviles producidas. Si el sistema se inicia con 5 unidades de coches reciclados, se irá incrementando la productividad ciclo a ciclo, hasta que el sistema se estabilice en 8 unidades de coches producidas y, por tanto, 6 unidades recicladas. Lo que significa que la producción de acero aumenta a 16 unidades, es decir, 10 que provienen de las 10 unidades de mineral y 6 que son producto de los coches reciclados.

Cuarta propiedad: la diversidad

La diversidad es otra de las características que definen los sistemas adaptables complejos. En cualquier sistema adaptable complejo existe una gran diversidad de agentes que, tras su coordinación, forman los patrones de comportamiento del sistema. A modo de ejemplo, podemos decir que en un bosque tropical es posible caminar durante medio kilómetro sin llegar a toparse dos veces con la misma especie de árbol. Pero no sólo en la selva tropical existe una diversidad tan grande: fijémonos ahora en el sistema adaptable complejo que podría suponer una ciudad como Roma, con millones de personas diferentes, cada una con su oficio y sus particularidades, y miles de comercios y empresas, cada uno, en su mayor parte, completamente distinto del otro, y a su vez, cada uno de estos negocios vuelve a ser un sistema adaptable complejo.

Esta diversidad no es accidental ni aleatoria. Cada agente dentro de un sistema ocupa su nicho de actuación, que básicamente viene definido por las conexiones que se han establecido con los agentes vecinos. Si se elimina un agente de un sistema adaptable complejo, el sistema se adapta para que automáticamente otros agentes ocupen el «agujero» dejado por aquél. Cuando el sistema ya ha dejado de adaptarse y llega a una situación estable, se dice que ha convergido.

La diversidad también aparece cuando un agente o conjunto de ellos se expanden hacia nuevos nichos de actuación, dando la oportunidad para la creación de nuevas funcionalidades que pueden ser explotadas por el sistema adaptable complejo. Un buen ejemplo de ello lo constituye el proceso de mimetismo, según el cual, y a modo de ejemplo, una orquídea evoluciona para que sus flores imiten la forma de un insecto con el fin de engañar a otros insectos, atraerlos y que éstos trasmitan más eficientemente el polen de una planta a otra.

Ophrys apifera *u orquídea abeja es una planta que imita formas de insectos en sus flores para atraerlos (fuente: Hans Hillewaert).*

Pero la principal pregunta que suelen plantearse los investigadores es: ¿qué es lo que permite e incluso motiva a un sistema adaptable complejo crear tanta diversidad? Pues bien, normalmente, cuando se plantea el estudio profundo de uno de estos sistemas se puede trazar paso a paso qué adaptaciones ha ido sufriendo para que emergiera un agente u otro, y la necesidad de cada uno de ellos se esclarece. Por ejemplo, en un escenario donde el sistema adaptable complejo se ha adaptado

para crear flujos cíclicos y, por tanto, para reciclar y ser más eficiente en lo global, se abren nichos que dan pie a la aparición de nuevos agentes, como los «agentes recicladores». Otro escenario que da pie a la diversidad es una empresa en crecimiento: es un sistema que necesita la aparición de nuevas jerarquías, de modo que requerirá la presencia de otro tipo de agentes que ejerzan la coordinación de cada nivel jerárquico.

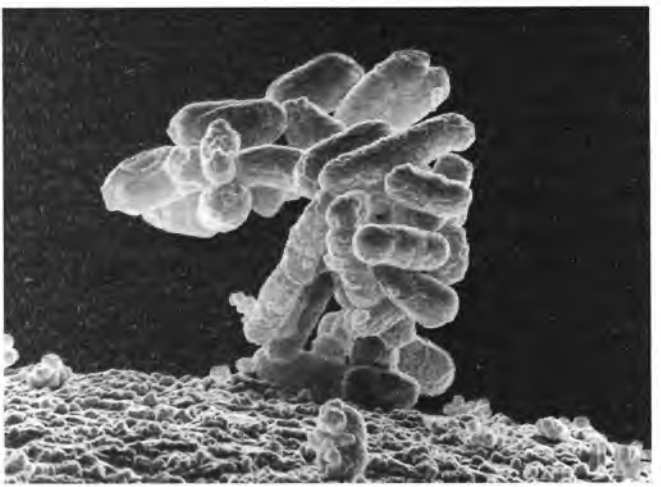
Segundo mecanismo: los modelados internos

Cada sistema adaptable complejo tiene cierta capacidad de crearse un modelo interno sobre el medio que le rodea, que proporciona, ante todo, una visión sobre futuros acontecimientos y los cambios que se deben ejecutar para poder adaptarse a ellos con éxito. Estos modelos sobre el medio son construidos a partir de los flujos de información que recibe el sistema y, a continuación, dichos flujos son transformados en cambios internos útiles que conforman los modelos. Una vez el modelo está construido, éste ayuda al sistema a poder anticipar las consecuencias que siguen cuando un determinado patrón aparece en el entorno. Pero un sistema ¿cómo puede trasformar la experiencia en modelos? ¿Cómo puede desarrollar un modelo para anticipar las consecuencias de eventos futuros?

Como siempre en la naturaleza, la presión evolutiva es la mejor herramienta para construir este tipo de mecanismos. El hecho de que una bacteria sepa que siempre debe seguir en la dirección marcada por el máximo gradiente de alimento es un «instinto» marcado por un modelo interno que le informa que si sigue ese patrón de comportamiento maximiza las posibilidades de asegurarse la fuente de alimento. Si una bacteria no ha llegado a crearse el modelo interno que le proporciona dicho instinto, tiene menos posibilidades de reproducirse y, por tanto, de dejar descendencia. Las bacterias que sí hayan codificado las estructuras y jerarquías entre sus agentes internos que le proporcionen la capacidad de crear ese modelo interno tendrán más posibilidades de reproducirse y, por tanto, de expandir esa propiedad al resto de la población.

Existen dos tipos diferentes de modelos internos, los implícitos y los explícitos. El ejemplo de la bacteria que sigue su instinto en busca de más alimento es un modelo implícito, ya que es un modelo que no le permite «pensar» ni simular qué pasaría si hiciera cualquier otra cosa. En cambio, un modelo explícito, que aparece en la naturaleza en entidades superiores, es una herramienta que sí permite a su poseedor tener una visión de varios escenarios hipotéticos, lo que le permite tomar

la mejor decisión después de analizar las diversas alternativas. Un ejemplo de modelado interno en un sistema adaptable complejo informatizado podría ser una máquina jugadora de ajedrez, capaz de analizar centenares de miles de movimientos en cada turno antes de mover la pieza. Lógicamente, cuando el modelo es implícito, se crea y se adapta al medio a una escala evolutiva, mientras que si es explícito, la velocidad de adaptación es mucho mayor.



Conjunto de bacterias de la especie Escherichia coli *ampliado 10.000 veces. Cada «bastoncito» corresponde a un individuo.*

Tercer mecanismo: los bloques de construcción

Un modelo interno de un sistema adaptable complejo está basado normalmente en un conjunto de muestras limitadas de situaciones pasadas que, por una parte, son similares, si bien cada una de ellas incorpora cierta novedad con respecto a la anterior. Pero ¿cómo puede un sistema adaptable complejo crear modelos internos en base a experiencias pasadas limitadas y que éstos sean útiles para nuevas situaciones futuras? La clave de esta paradoja reside en el uso de los denominados bloques de construcción. Se trata de los elementos en los que todo sistema, entorno o escenario se puede descomponer. Por ejemplo, examinemos el caso de una entidad financiera que, por otra parte, responde perfectamente a la definición de sistema adaptable complejo. Supongamos que la entidad está tratando de dilucidar si concede un

préstamo a un nuevo cliente o no, y su preocupación principal es saber, lógicamente, si el cliente en cuestión será capaz de devolverlo en el plazo acordado. El banco no tiene ni idea de si el cliente podrá continuar pagando las cuotas del crédito dentro de 15 años, ya que no puede adivinar el futuro. Supongamos, para mayor dificultad, que además es un cliente completamente nuevo, sin historial crediticio y, por tanto, sin ninguna referencia previa. Lo que haría el banco en este caso es descomponer el problema y, analizando las características que definen el nuevo cliente, como nivel formativo, oficio, estado civil, etc., ver cómo se han comportado clientes que responden a su mismo perfil. Para un banco que está analizando un nuevo cliente, esas características son los bloques de construcción que definen el escenario ante el que se encuentra este sistema adaptable complejo.

La capacidad de combinar bloques de construcción para conformar los modelos internos implícitos se realiza a escala evolutiva, mientras que el aprendizaje en modelos internos explícitos es una capacidad que suele ejecutarse a escalas mucho más reducidas, aunque en la naturaleza sólo se da en animales superiores.

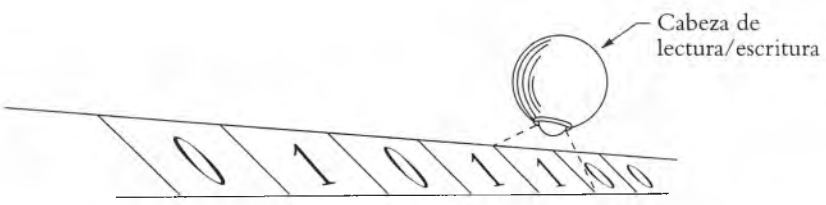
Los autómatas celulares

El ejemplo más clásico de vida artificial (o, mejor dicho, sistema adaptable complejo) en el campo de la informática es el de los autómatas celulares. Se trata de un concepto bastante simple que ayuda a explorar la complejidad de sistemas superiores, fruto de la investigación de dos de los matemáticos dedicados al mundo de la computación más reputados, Stanislaw Ulam (1909-1984) y John von Neumann (1903-1957), a los que además unía una gran amistad.

El matemático estadounidense de origen polaco Stanislaw Ulam.

Los autómatas, en general, son formulismos matemáticos que, ante una determinada entrada, ejecutan una serie de instrucciones programadas previamente. Por decirlo en otras palabras, un autómata es una generalización de un algoritmo o un programa informático. Por eso, en informática todo son autómatas, desde un microchip programado para realizar unas determinadas acciones hasta un sistema operativo. Un ejemplo de autómata que ya se vio en el primer capítulo es la máquina de Turing.

Normalmente, los autómatas teóricos, como la máquina de Turing, son instrumentos que recogen sus entradas e imprimen sus salidas en cintas unidimensionales. De este modo, el autómata va viajando por encima de la cinta, a izquierda o derecha, leyendo los símbolos que hay escritos en ella, tal como se muestra en la figura siguiente; a partir de ellos y su programación, realiza una acción u otra, como podría ser imprimir un determinado símbolo en una parte de la cinta.



Dos de los componentes fundamentales de una máquina de Turing: la cinta de papel y la cabeza lectora con capacidad de escritura (fuente: Complexity, *de Melanie Mitchell).*

Sin embargo, los autómatas celulares son una clase particular de autómatas que no viajan por encima de cintas bidimensionales, sino que en ellos el medio de entrada/salida es un casillero plano, como un tablero similar al de ajedrez, y en cada una de las casillas hay un autómata celular que no se desplaza. En los autómatas celulares las entradas de información son las casillas adyacentes a aquella en la que está situado el autómata, y la salida de información se realiza en la propia casilla donde está ubicado.

Cada autómata dispuesto en cada una de las casillas del tablero tiene programadas una serie de instrucciones. Por ejemplo, si el número de casillas negras que rodean la casilla en la que está situado el autómata celular es par, pinta la casilla de salida de negro y, si no, la pinta de blanco. En este caso, y ubicando un autómata celular en cada una de las casillas del tablero, éste puede ir adoptando diversas for-

mas o dibujos, cambiantes en función de lo que los diferentes autómatas celulares estén pintando en ese momento en las casillas.

Entre las infinitas configuraciones que puede presentar un autómata celular, hay un conjunto de ellas que dan pie a la emergencia de eventos perpetuos, como es el caso del automatismo de Conway o juego de la vida. De hecho, en Internet el lector puede encontrar un gran número de configuraciones que dan lugar a la emergencia de graciosos dibujos que se crean, se autodestruyen y se vuelven a crear, y todo ello programado con normas realmente simples, similares a las del automatismo de Conway.

Dibujo de una máquina de Turing a vapor pintada por los estudiantes de la Universidad de Washington en una de las salas de la universidad.

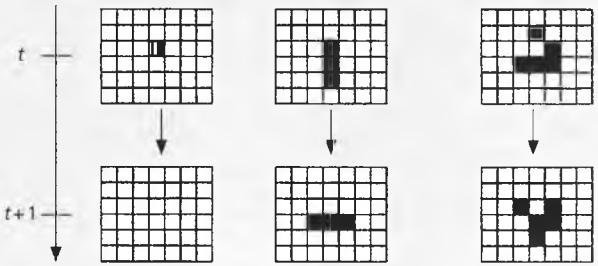
EL AUTOMATISMO DE CONWAY O EL JUEGO DE LA VIDA

El juego de la vida, ideado por John Horton Conway (n. 1937), es una programación de autómatas celulares que, a pesar de su simplicidad, da lugar a la aparición de un fascinante comportamiento emergente. Las reglas son simplemente dos, y se deben tener en cuenta las ocho casillas que rodean a cada una de las demás, más el propio estado de la casilla en la que se ubica el autómata celular:

Regla 1. Si el color de la casilla es «blanco» y exactamente tres casillas vecinas tienen color «negro», entonces el color de la casilla cambia a «negro»; si no, continúa «blanco».

Regla 2. Si el color de la casilla es «negro» y dos o tres vecinos también tienen el color «negro», entonces la casilla continúa de color «negro»; si no, cambia a «blanco».

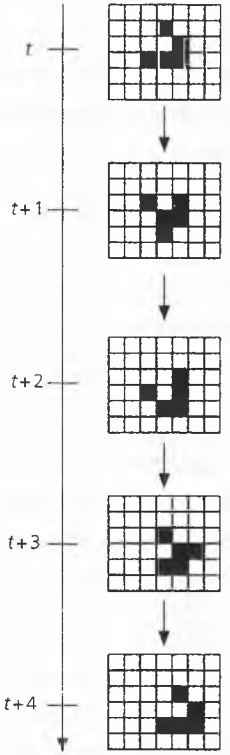
Si el lector tiene conocimientos básicos de programación de ordenadores, se recomienda que implemente estas sencillas normas para ver el comportamiento en vivo. En caso contrario, a continuación se muestras algunos ejemplos de comportamiento:



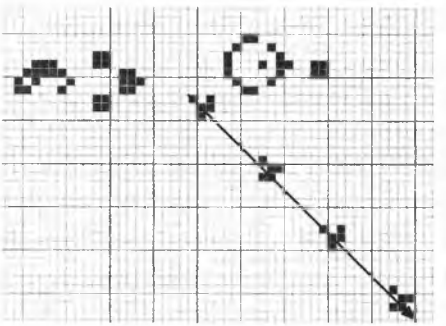
Es justamente el ejemplo emergente que surge al programar las reglas del juego de la vida lo que se conoce como «glider», que sería la siguiente sucesión cíclica:

Sistemas inmunes artificiales

Imitar el comportamiento inteligente y sabio de la naturaleza siempre ha sido una gran fuente de inspiración para los ingenieros especializados en inteligencia artificial. La naturaleza inspiró en su momento la idea de las redes neuronales y los algoritmos evolutivos ya explicados anteriormente, de gran importancia en la historia de la inteligencia artificial. De la misma fuente de inspiración han surgido otras técnicas, como los sistemas inmunes artificiales, que tratan de imitar el comportamiento del sistema inmune animal, o la inteligencia de enjambre, que intenta imitar el comportamiento individual y simple de cada uno de los miembros de una colo-



Como puede apreciarse en la figura de la izquierda, la forma en $t+4$ es idéntica a la de t, pero toda ella se ha desplazado un cuadro hacia abajo y otro hacia la derecha. Por tanto, si repetimos las operaciones hasta $t+9$, veremos que el «glider» (que viene a ser el nombre que recibe el dibujo) se vuelve a desplazar por el tablero en la dirección diagonal que se muestra en la figura inferior:



Versión más sofisticada del «glider». Si la imagen estuviera animada, veríamos cómo los dibujos situados bajo la flecha se desplazan en la dirección a la que ésta apunta.

nia (por ejemplo, un enjambre de abejas) para, de manera agregada, simular ciertos patrones de comportamiento aparentemente inteligentes.

El sistema inmune de un animal es, en cierto modo, un sistema de optimización y reconocimiento de patrones muy eficiente, ya que, dado un nuevo problema que hay que resolver (un antígeno que se ha introducido en el cuerpo), rápidamente diseña, mediante un ordenado proceso de prueba y error, la solución a dicho problema, o dicho en términos biológicos, el anticuerpo que reconoce al antígeno en cuestión.

El funcionamiento del sistema inmune es bastante similar al de un proceso evolutivo, con la gran salvedad de que en este caso no se cruzan las diferentes propues-

tas de soluciones para tratar de identificar una solución promedio que combine las bondades de los progenitores. El procedimiento puede resumirse de la siguiente manera:

1. Se genera una gran variedad de anticuerpos de manera aleatoria.
2. Se evalúa la bondad de cada uno de esos anticuerpos, es decir, se analiza si puede reconocer al antígeno que está atacando el cuerpo.
3. A partir de ellos se crea una segunda generación de anticuerpos según la siguiente estrategia:
    a) Se replican los anticuerpos en múltiples copias. Cada anticuerpo es multirreplicado proporcionalmente a su bondad, es decir, un anticuerpo muy eficaz será replicado muchas veces para la nueva generación, mientras que uno malo, o bien no es replicado o bien es replicado muy pocas veces.
    b) Se introducen variaciones en las copias de los anticuerpos (o mutaciones si usamos la terminología de los algoritmos evolutivos), de manera inversamente proporcional a su eficacia, es decir, las copias de los anticuerpos buenos prácticamente no serán modificadas en la nueva población (pero algo sí), mientras que las réplicas de los malos anticuerpos sufrirán grandes variaciones.
4. Los nuevos anticuerpos diseñados en los pasos anteriores son de nuevo evaluados ante el antígeno, y el proceso se repite para crear una nueva generación de anticuerpos.
5. Cuando el sistema biológico considera que ya tiene un buen anticuerpo eficiente para reconocer el antígeno, el proceso se detiene.

Imitar el proceso que sigue el sistema inmune para diseñar anticuerpos se puede adaptar fácilmente a la resolución de problemas reales. De hecho, el único paso crítico es cómo representar las posibles soluciones a un problema para que éstas puedan ser replicadas y variadas. Lo que se recomienda en este caso es seguir la metodología usada en los algoritmos evolutivos, es decir, codificar la solución mediante cromosomas compuestos a partir de genes. Aunque estemos mezclando terminología de dos técnicas aparentemente muy diferentes, el invento funciona y cada vez se usan más sistemas inmunes artificiales para resolver problemas reales de ingeniería, no sólo por su eficacia en la optimización, sino también porque se adaptan muy bien a la arquitectura de los modernos esquemas de supercomputación y computación distribuida

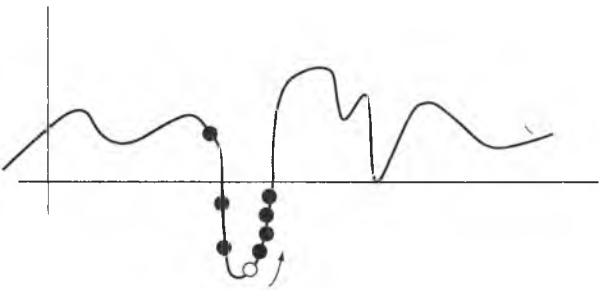
basados en *grid-computing* (del inglés, «computación en malla») y *cloud-computing* («computación en la nube»). En estos casos, la potencia de cálculo está distribuida en una «nube» abstracta y difusa de ordenadores que individualmente son muy potentes, aunque no existe necesariamente una buena comunicación entre ellos. Por eso, el control central de un sistema inmune puede mandar evaluar los anticuerpos a la nube, y cuando se termina la evaluación, el sistema central diseña la siguiente generación. En este escenario, es en la evaluación individual de los anticuerpos donde existe la mayor parte del coste computacional, y por eso se mandan evaluar a la nube de cálculo, mientras que la creación de las nuevas generaciones se puede realizar de manera secuencial con poco coste en el sistema central.

Inteligencia de enjambre

La inteligencia de enjambre (en inglés *swarm intelligence*) vuelve a estar inspirada en la naturaleza. El término fue introducido por Gerardo Beni y Jing Wang a finales de la década de 1980. La inteligencia de enjambre está basada en simular el comportamiento individual de entidades simples de manera que, al agregar el comportamiento de muchas entidades iguales, emerja un comportamiento global que pueda tener cierta inteligencia. Por ello, el principal reto en la implementación de un sistema de inteligencia de enjambre es definir cómo interacciona cada entidad con su entidad vecina y con el medio. A partir de esta política, si está bien definida, al agregar la actividad de todas las entidades de la colonia o del enjambre deberá emerger un comportamiento inteligente global.

Veamos un caso práctico inspirado en el comportamiento de una bandada de pájaros que surca los cielos. Imaginemos que se desea encontrar el óptimo de una función matemática compleja con centenares de dimensiones, con diversos máximos y mínimos locales, etc. Ahora, para empezar (diremos que $t = 0$) situamos cien «pájaros» de manera aleatoria, pero agrupada, en una zona de la función. Cada uno de ellos, de manera individual, cada vez que avanzamos en la línea del tiempo ($t' = t + 1$) sólo debe tener en cuenta dos cosas: a) la dirección (A) en la que está situado el «centro de masas» de la bandada, es decir, la dirección promedio en la que se encuentra el resto de sus compañeros, para no separarse en exceso de ellos, y b) la dirección (B) hacia la que se dirige el máximo gradiente de la función que hay que optimizar; es decir, como queremos encontrar el máximo de la función, hace falta saber hacia qué dirección crece más rápidamente la función. A partir de las dos direcciones calculadas, A y B, se calcula una tercera, $C = A + B$, y cada «pá-

jaro» debe desplazarse ligeramente en esa dirección C. Como todos los «pájaros» están reproduciendo estas reglas de desplazamiento, la bandada va navegando por la función, sin separarse excesivamente y buscando el máximo global. La ventaja de utilizar un grupo de «pájaros» y no uno solo es que, al usar varios puntos de exploración (cada uno de los «pájaros»), la superficie de muestreo de la función es mayor y se reduce la posibilidad de caer en máximos locales, lejos del máximo global.



En la figura anterior, los puntos negros representan los diversos «pájaros» de la bandada, y el punto blanco, el centro de masas de la misma. La flecha muestra la dirección global que seguirá la bandada en búsqueda del máximo global.

La inteligencia de enjambre se inspira en el modo de desplazarse de algunos pájaros, como los estorninos, que se agrupan en enormes bandadas que dibujan unas formas muy curiosas en el cielo.

Sin embargo, a pesar de la innovación que supone la inteligencia de enjambre, el uso de estos métodos en la resolución de problemas reales es aún incipiente. En la actualidad existen dos ámbitos altamente relacionados donde estas técnicas están siendo exploradas de forma intensiva para el control y la navegación automática de vehículos: el sector aeroespacial y el sector militar.

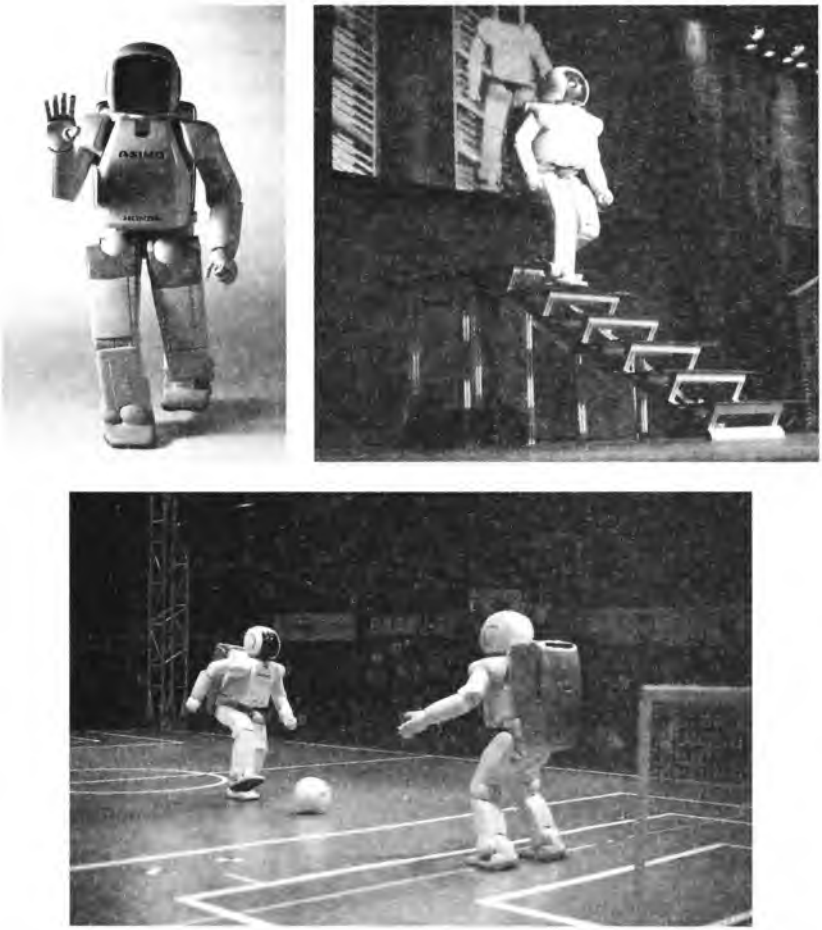
Aplicaciones de la vida artificial

La vida artificial es un campo relativamente nuevo en el ámbito de la inteligencia artificial. Es por esto que aún son incipientes las aplicaciones que se le están dando. Sin embargo, en el futuro, complejas tareas de control, supervisión y planificación serán llevadas a cabo por sistemas «vivos», como ya está pasando en el caso de las inversiones especulativas bursátiles.

Teoría de juegos

La teoría de juegos es una rama de las matemáticas que se dedica a estudiar las interacciones entre estructuras de incentivos y cómo llevar a cabo procesos de decisión. El objetivo final es identificar las estrategias óptimas y predecir el comportamiento de los individuos involucrados en una de esas estructuras ante una situación concreta. Los matemáticos John von Neumann y Oskar Morgenstern fundaron las bases de esta disciplina durante la Guerra Fría con el fin de identificar las estrategias militares óptimas, aunque rápidamente fue expandiendo su aplicación a la economía, la política, la ética, la filosofía, la biología y, por supuesto, la teoría de la computación.

La teoría de juegos es de mucha utilidad en el estudio de los sistemas adaptables complejos, ya que a menudo los agentes que componen dichos sistemas deben competir o cooperar entre ellos para conseguir el bien global del sistema. A menudo, en un esquema cooperativo el esfuerzo individual de un agente es mayor que el beneficio global repartido de manera proporcional entre todos los agentes que componen el sistema. Sin embargo, este esfuerzo puede ser imprescindible para alcanzar el bien global, cuyo beneficio, en valor absoluto, puede multiplicar por varios órdenes de magnitud el esfuerzo individual. Así pues, para incentivar el comportamiento adecuado de los agentes que componen el sistema y poder predecir la viabilidad de un sistema adaptable complejo a través del comportamiento de aquéllos, deben usarse análisis de teoría de juegos.



El famoso robot de Honda bautizado como ASIMO. Es capaz de realizar actividades tan humanas como bajar escaleras o jugar a fútbol.

De nuevo el *data mining*

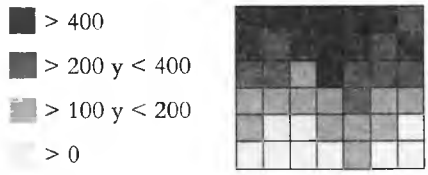
La vida artificial es un concepto evocador y un tanto misterioso para los profanos. Sin embargo, los conceptos vistos hasta el momento que encajan dentro de la definición de vida artificial, como los autómatas celulares, son usados en ingeniería para tareas bastante prosaicas. Una de esas aplicaciones es el análisis inteligente de datos,

o *data mining*, que ya hemos visto con anterioridad. En un problema de análisis de datos hace falta procesar grandes cantidades de datos para extraer conclusiones a partir de ellos, pero los volúmenes de información que deben procesarse normalmente tienen unas dimensiones tan grandes que son casi imposibles de manejar por expertos humanos. Por eso suelen utilizarse herramientas informatizadas inteligentes con el fin de desarrollar análisis de tendencias entre los datos.

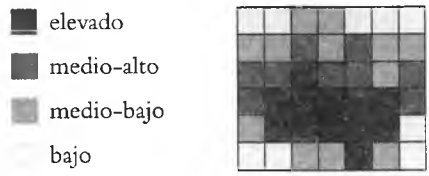
A pesar de que el análisis de datos se puede llevar a cabo con una gran variedad de herramientas, los autómatas celulares aportan algo distinto y ese algo es la capacidad de interrelacionar los datos espacialmente. Por ejemplo, imaginemos que estamos analizando los datos de las ventas de paraguas en un país concreto. Los datos de las ventas, desglosados por clientes, pueden ser procesados sin tener en cuenta la distribución espacial de las ventas, o como mucho, introduciendo la distribución espacial como una variable categórica, es decir, el cliente A compró 20 unidades y A es de la ciudad X, mientras que B compró 240 unidades y es de la ciudad Y, y C compró 4.530 unidades y es de Z. En un sistema que no pueda tener en cuenta la distribución espacial, las ciudades X,Y y Z no son más que categorías, y difícilmente se puede tener en cuenta que X está a 150 km al sur de Y, y que Y está a 400 km al sur de Z. Si se tuviera en cuenta ese dato, se vería que en ese determinado país, la región del norte es la más lluviosa, y por eso, a medida que se viaja hacia el sur, disminuyen drásticamente las ventas de paraguas.

Ahora bien, si estos datos geográficos los representamos encima de un tablero (como lo hacen los autómatas celulares), de manera que la distribución espacial tenga una cierta relación con la distribución geográfica real de la procedencia de los datos, el análisis que se puede desarrollar de esta información sí que tendrá en cuenta la distribución espacial de manera más inteligente que una simple categorización.

Por ese motivo, una vez que los datos están distribuidos en un casillero, se puede usar una metodología de tipo algoritmo evolutivo para que encuentre, por evolución, las reglas que el autómata celular debe implementar para ejecutar el análisis de datos. Si volvemos al ejemplo de la venta de paraguas e introducimos como dato asociado a las ventas la pluviometría de cada uno de los puntos de venta, podemos hacer que un algoritmo nos evolucione un conjunto de reglas cuya salida sea pintar la casilla de un color u otro en función de las ventas de paraguas de cada uno de los puntos de venta, descontando el efecto pluviométrico. De este modo, si pintáramos el mapa de ventas sin tener en cuenta la pluviometría, tendríamos un mapa como el siguiente:



Sin embargo, al eliminar el efecto introducido por la pluviometría de los lugares, podría adoptar el siguiente aspecto:



Todo ello puede indicarle a un experto que en la zona centro-sur del mapa es donde se venden más paraguas, y esto muestra la tendencia de que en aquella parte del país hay un mayor poder adquisitivo, ya que, dadas las características de la región, la población es capaz de gastar su dinero en un producto que no le es del todo necesario. Lo que el distribuidor de paraguas haría a continuación sería aumentar el precio de los paraguas en la zona centro-sur del país, ya que, aunque se venden menos unidades, la gente los compra por lujo más que por necesidad y, por tanto, es menos sensible al precio.

Programación de robots

Otro aspecto muy importante para el que es útil la vida artificial y, en general, los sistemas adaptables complejos, es para la programación del comportamiento de un robot. Cada vez más están llegando al gran consumo robots domésticos capaces de realizar tareas sencillas de limpieza, como puede ser aspirar o fregar el suelo, o detectar la presencia de intrusos. Estos robots suelen tener una cierta movilidad, ya que se desplazan dentro de los hogares mediante ruedas, pero deben ser guiados por un sistema inteligente para saber hacia qué dirección deben ir y qué acción deben llevar a cabo en cada ubicación. En el caso más simple, el de los robots-aspiradora, vamos a ver cómo éstos ya cumplen con las definiciones de los sistemas adaptables complejos:

El robot aspiradora es uno de los robots domésticos más conocidos.

— Agregación. Por supuesto, estos robots son sistemas agregados, ya que tienen motores, detectores de presencia, la propia aspiradora, la unidad de procesamiento que determina la dirección hacia la que dirigirse, etc.

— Etiquetado. Estos robots pueden etiquetar e interaccionar con un medio etiquetado. Por ejemplo, si detectan que una zona del espacio acumula más suciedad de lo normal, la etiquetan como tal y en ella focalizan un mayor esfuerzo. También el usuario puede etiquetar una zona por la que el robot no debe pasar y éste puede detectarla y evitarla.

— No-linealidad. De nuevo el comportamiento de estos robots es claramente no-lineal, ya que la suma de las partes es capaz de realizar tareas de un valor superior que la suma individual de las partes. Un motor, unas ruedas, una aspiradora, etc., de manera independiente, no pueden mantener limpio de polvo un hogar sin intervención humana, mientras que sí tienen esa capacidad cuando están todos coordinados dentro del sistema adaptable complejo que supone uno de estos robots.

— Flujos. El mismo robot es un complejo sistema de gestión de flujos de información, de modo que ésta fluye entre el medio y las diferentes partes del robot. Éste cuenta con un determinado número de sensores que le suministran datos sobre el medio, como, por ejemplo, dónde se encuentran las paredes o si está entrando en una zona particularmente polvorienta. Esta información circula hasta la unidad central de procesamiento y allí se analiza y se envían otras señales a los diferentes motores del robot, que ejecutan órdenes que modifican el medio original desde donde fluyeron las señales iniciales.

En el caso de encontrar suciedad, se ordenará aumentar la potencia de aspirado, y si se encuentra con un obstáculo se puede ordenar dar media vuelta.

— Diversidad. También reflejan diversidad en su modo de comportarse, de modo que si encuentran un obstáculo buscan la forma de sortearlo. La diversidad radica en que no sortean los obstáculos siempre del mismo modo, sino que van alternando modos de sorteo con el fin de minimizar las posibilidades de quedar atrapado infinitamente en un mismo lugar.

— Modelos internos. Disponen de unos modelos internos implícitos, de modo que al inicio de su funcionamiento van siguiendo una ruta arbitraria, pero a medida que van conociendo más superficie explorada, van concentrando su acción en aquellas zonas donde se acumula más polvo.

— Bloques de construcción. Finalmente, estos robots usan bloques de construcción en sus modelos internos. Por ejemplo, si encuentran una pared, tratan de sortear el obstáculo, hasta que se dan cuenta de que es una pared y que ese obstáculo es insorteable; entonces toman otra estrategia. Lógicamente, los robots aspiradora no están entrenados para adoptar ese comportamiento en todos los tipos de paredes o elementos parecidos a una pared, sino que se usa el bloque de construcción «pared», y cuando éste aparece, se ejecuta la estrategia.

Epílogo

Predecir es difícil, especialmente cuando se trata del futuro.
Niels Bohr

Llegados a este punto, el lector ya debería tener una ligera idea de qué hay de realidad y qué hay de invención en todos los cuentos, novelas y películas de ciencia ficción que, demasiado a menudo, nos muestran máquinas pensantes casi humanas.

Hoy en día la inteligencia artificial no es más que un conjunto de herramientas avanzadas que permiten solucionar complejos problemas de manera rápida, aproximada y basándose en la experiencia adquirida, tal y como lo haría un ser humano. A veces, podemos disfrutar de cierta creatividad o innovación en estos algoritmos que la ciencia y la tecnología moderna han puesto a nuestro alcance, pero desgraciadamente aún estamos lejos de encontrarnos con máquinas, incluso en los ámbitos de la investigación más puntera, que puedan ser confundidas con un ser humano. Aun así, diariamente y de manera inconsciente vamos interactuando con verdaderas herramientas, sistemas y aparatos que implementan algoritmos inteligentes, como pueden ser los sistemas de conducción de muchos trenes europeos, la depuración de aguas residuales de la mayoría de las grandes ciudades o la planificación logística del movimiento de miles de millones de toneladas de mercancías que a diario recorren nuestro planeta.

De hecho, poco a poco, la inteligencia artificial va penetrando en nuestras vidas y, tarde o temprano, llegará el día en que podremos interactuar con máquinas con el mismo nivel de creatividad, percepción e inteligencia emocional que un ser humano; entonces, esta vez sí, sabremos que no estamos solos.

Anexo

Conversación con Eliza

En el capítulo 1 ya tuvimos una pequeña conversación con Eliza, el «psicólogo automático» desarrollado por Joseph Weizenbaum. En este anexo me he esforzado en tener otra conversación real un poco más larga para que el lector pueda observar su comportamiento en un entorno un poco más extenso. Por supuesto, los motivos que se le dan a Eliza son completamente falsos y sirven sólo para ilustrar su comportamiento:

Eliza: Hola. ¿Cómo estás? Cuéntame tu problema.
I. Belda: Estoy estresado.
Eliza: ¿Desde cuándo estás estresado?
I. Belda: Desde que mi editor me dijo que tenía que escribir más rápido este libro.
Eliza: ¿Y por qué te dijo tu editor que escribieras rápido este libro?
I. Belda: Porque quiere que lo entregue antes de enero.
Eliza: ¿Y es ésa una razón real?
I. Belda: Creo que él también está presionado.
Eliza: ¿Puedes explicármelo mejor?
I. Belda: Me dijo que necesitaban bastantes títulos para publicarlos inmediatamente.
Eliza: ¿Y eso explica el hecho de que quiera que escribas ese libro rápido?
I. Belda: ¿Tú qué crees?
Eliza: ¿Por qué me lo preguntas?
I. Belda: Porque es obvio.
Eliza: ¿Y qué otras razones te vienen a la cabeza?
I. Belda: Creo que otros escritores se están demorando.
Eliza: ¿Y dices que otros escritores se están demorando por alguna razón?
I. Belda: Porque él me lo dijo.
Eliza: ¿Te parece que esto explica alguna otra cosa?
I. Belda: No, simplemente que él también está bajo presión.
Eliza: Estás siendo un poco negativo.

I. Belda: ¿Por qué lo dices?
Eliza: ¿Este tema te interesa?
I. Belda: Sí, me interesaría saber por qué crees que soy un poco negativo.
Eliza: Ya veo.

En este punto el autor se cansó de continuar dando coba a Eliza y dejó la conversación, pero el lector ya habrá observado que su comportamiento es en realidad muy simple. Lo primero que Eliza intenta hacer ante cualquier afirmación es preguntar por las razones, luego por las razones de las razones, y así sucesivamente. Cuando se llega a un callejón sin salida, intenta salirse por la tangente con frases como «Estás siendo un poco negativo» o «Ya veo».

Glosario

Algoritmo evolutivo. Método de búsqueda y optimización basado en la evolución natural. En un algoritmo evolutivo se proponen distintas soluciones posibles a un problema, éstas se evalúan y las mejores compiten entre sí para acabar obteniendo la solución óptima.

Algoritmo genético. Clase particular de algoritmo evolutivo. En general, en los algoritmos genéticos las soluciones a un determinado problema se codifican mediante una secuencia de bits. Las secuencias (llamadas genes) que representan las mejores soluciones (o individuos) son cruzadas entre sí y mutadas, simulando al máximo el proceso evolutivo biológico. El algoritmo genético fue uno de los primeros esquemas evolutivos que popularizaron estas técnicas inteligentes.

Análisis de componentes principales. Normalmente abreviada por sus siglas en inglés, PCA, es una popular herramienta estadística usada para determinar las componentes, o variables, que mayor variabilidad introducen en los datos estudiados.

Árbol de decisión. Herramienta informática usada para clasificar muestras estadísticas. La clasificación se basa en el análisis de las componentes más decisivas o discriminantes que decantan una muestra hacia una clase u otra. Se trata de una herramienta muy simple, pero también muy efectiva, para el reconocimiento de patrones.

Autómata. Máquina, que no necesariamente tiene que tener una representación física, que puede ser programada para que responda de determinadas maneras a las entradas que recibe.

Autómata celular. Caso particular de autómata programable y el ejemplo más simple de vida artificial. Un autómata celular tiene un comportamiento espacial, es decir, recibe las entradas de su área adyacente y, en función de la situación de su entorno, adopta un comportamiento u otro.

Clustering. Consiste en agrupar muestras estadísticas atendiendo a criterios varios. El reto de las herramientas de *clustering* es detectar los criterios de agrupación de manera inteligente. Tiene múltiples aplicaciones en todas las disciplinas científicas.

Computación evolutiva. Disciplina que estudia los algoritmos evolutivos, su configuración óptima y su aplicación a la resolución de problemas. *Véase* algoritmo evolutivo.

Computador universal. Herramienta que puede computar cualquier algoritmo. Un computador universal es una entelequia matemática usada para comprobar que un nuevo lenguaje de programación o un nuevo dispositivo electrónico puede implementar todas las funciones para las que será usado.

Diversidad. Concepto estudiado en la computación evolutiva para determinar la variancia genética de la población (conjunto de soluciones propuestas) de un algoritmo evolutivo y cómo evolucionan éstas a lo largo del tiempo. El estudio de la diversidad genética en una evolución es crucial para determinar la configuración óptima del algoritmo y que no conduzca a subóptimos locales.

Entropía de Shannon. Concepto matemático muy usado en telecomunicaciones para determinar el «desorden» o la entropía de una señal. Básicamente es una medición sobre el número de símbolos distintos y su frecuencia de aparición en una señal o en una fuente de datos. La entropía de Shannon también es usada en criptografía y en la compresión de datos.

Inteligencia de enjambre. Sistema complejo de vida artificial usado en la resolución de determinados problemas. La inteligencia de enjambre trata de programar autómatas con una «inteligencia» muy simple. Al agregar decenas o centenares de dichos autómatas, la inteligencia global aumenta de manera no lineal hasta alcanzar una inteligencia de grupo significativa.

Lógica booleana. Es una lógica matemática basada en el álgebra de Boole, donde las variables sólo pueden adoptar los valores de «verdadero» o «falso». Toda la electrónica digital moderna se fundamenta en la lógica booleana, con excepción de los últimos avances en la computación cuántica.

Máquina de soporte vectorial. Potente y popular herramienta matemática introducida por el científico Vladimir Vapnik a principios del siglo XXI capaz de clasificar muestras estadísticas mediante la introducción de nuevas dimensiones «artificiales» en las variables de un problema. El nombre de la herramienta proviene del hecho de que, para clasificar los datos estadísticos, se determinan los vectores o muestras estadísticas de clasificación conocida que determinan los soportes del hiperplano que mejor separan las muestras de distinta clase.

Máquina de Turing. Caso particular de autómata programable que recibe su entrada desde una cinta infinita y tiene la capacidad de desplazarse y escribir sobre dicha cinta. Se supone que

una máquina de Turing es un computador universal, aunque esto aún no ha podido ser demostrado matemáticamente. La máquina de Turing es una entelequia matemática muy usada en la teoría de la computación, ya que se utiliza para verificar que un nuevo lenguaje de programación puede computar cualquier algoritmo, implementando con dicho lenguaje una máquina de Turing.

Minería de datos. Rama del análisis de datos que es capaz de extraer nuevo conocimiento e inferir reglas no evidentes a partir de un gran volumen de muestras. La minería de datos es capaz de establecer relaciones entre datos que se presentan en un volumen excesivo para que la mente humana pueda procesarlos y extraer hipótesis concluyentes.

Red bayesiana. Herramienta matemática basada en la concatenación de probabilidades de determinados eventos siguiendo la probabilidad condicionada descrita en el siglo XVIII por el reverendo Thomas Bayes. La idea subyacente a una red bayesiana es que hay cadenas de eventos que suelen sucederse, y que éstas pueden compartir probabilidades con otras cadenas de eventos; por este motivo se llaman redes, ya que son cadenas de probabilidades entrecruzadas.

Red neuronal. Herramienta matemática consistente en una red de neuronas artificiales capaces de ser entrenadas para resolver problemas de clasificación. Las redes neuronales imitan el comportamiento del sistema nervioso animal, también compuesto por neuronas entrenadas mediante un proceso de aprendizaje.

Sistema experto. Antiguo método inteligente consistente en crear programas informáticos expertos en una determinada disciplina técnica o científica. El razonamiento de estos programas estaba completamente circunscrito al conocimiento introducido en el momento de la programación y difícilmente podían llegar a aprender a partir de nuevas experiencias; ése es el motivo por el que dejaron de ser usados.

Sobreentrenamiento. Tiene lugar cuando un algoritmo de clasificación ha sido entrenado de manera que no es capaz de generalizar, sino solamente de memorizar. Cuando esto ocurre el algoritmo no puede clasificar correctamente nuevas muestras, ya que tan sólo puede hacerlo con aquellas que ha memorizado durante el entrenamiento. El sobreentrenamiento suele suceder cuando se somete a estos algoritmos a procesos de aprendizaje demasiado largos.

Variable latente. Variable estadística que describe a la vez varias condiciones de una muestra. Algunos ejemplos de variables latentes muy usados son variables como la «riqueza» de una sociedad o el bienestar de una población. Estas variables introducen una mayor densidad de información al condensar varias variables simples en una. Existen métodos automáticos de creación de variables latentes, como el análisis de componentes variables, que no sólo las crean sino que, además, escogen aquellas sobre las que existe la mayor variabilidad posible entre los datos.

Bibliografía

CASTI, J.L., *El quinteto de Cambridge*, Madrid, Taurus, 1998.

GOLDBERG, D.E., Genetic Algorithms in Search, Optimization, and Machine Learning, Boston, Addison-Wesley, 1989.

—: *The Design of Innovation. Lessons from and for Competent Genetic Algorithms*, Norwell, Kluwer Academic Publishers, 2002.

HOLLAND, J.H., *Adaptation in Natural and Artificial Systems*, Cambridge/Londres, MIT Press/Bradford Books, 1992.

—: *Emergence. From Chaos to Order*, Cambridge, Perseus Books, 1998.

—: *Hidden Order. How Adaptation Builds Complexity*, Reading, Perseus Books, 1995.

MCELREATH, R., ROBERT, B., *Mathematical Models of Social Evolution. A Guide for the Perplexed*, Chicago, The University of Chicago Press, 2007.

Índice analítico

agregación 116, 137
ajedrez 16, 17, 20, 28, 35, 124
álgebra booleana 30-31
algoritmos
  evolutivos 10, 39, 43, 46-49, 92, 143
  genéticos 42, 45-47, 143
análisis de componentes principales 66, 100, 101, 107, 143
árbol de decisión 75, 77, 143
asunción de mundo cerrado 34
autómata 126, 143, 144
  celular 125-128, 134, 135, 143
automatismo de Conway 127, 128

back-propagation 70
back-tracking 87, 88
Beni, Gerardo 131
bloques de construcción 124-128, 138
Boole, George 29, 30-31, 144
branch-and-bound 87-88

cinemática inversa 72-73
clustering 106, 143
computación evolutiva 38-39, 50, 143, 144
computador universal 15, 144, 145
Conway, John 127-129
CWA 34-35

De Morgan, Augustus 29
Deep Blue 16-17
diversidad 41, 44, 121-123, 137, 144
docking 51

ecuaciones de Lotka-Volterra 118-119
efecto de Hughes 99
Eliza 32-33, 141-142
entropía de Shannon 77, 144
etiquetado 116-117, 120, 137

feature selection 100
feed forward 69
flash crash 55-56
flujos 119-121, 123, 137

glider 128-129
go 20-21
greedy backward elimination 102
greedy forward selection 102

habitación china 16-18, 111
heurística 10-11, 27-30, 88
high frequency traders 56
Holland, John 39, 111

instancia 117
inteligencia de enjambre 128, 131-133, 144

juego de la vida 127-129

Kant, Immanuel 105
Kasparov, Gary 16-17

Logic Theorist 26-27
lógica booleana 31-32, 87, 144

maldición de la dimensionalidad 99-104
mamografía 56-60
mapa
  autoorganizativo 72
  cognitivo difuso 92
  de Kohonen 72
máquinas de soporte vectorial 74, 107
marketing on-line 61-64
McCorduck, Pamela 26
McCulloch, Warren 65
Microsoft Research 62, 110
minería de datos 95-99, 145
modelos internos 123-125, 138
Morgenstern, Oksar 133

neurona artificial 65-66
Newel, Alan 26, 28
no-free lunch, teorema de 88-89
no-linealidad 117-119, 137
no-separabilidad lineal 68-69

perceptrón 65-69
pigeon ranking 83
Pitts, Walter 65
Playfair, William 105
Priestley, Joseph 104

random forest 79
razonamiento basado en casos 57
red
  bayesiana 62, 145
  de Hopfield 71
  neuronal 9, 65, 69-74, 92, 97, 103, 145
Reiter, Raymond 34
restricciones 18, 25, 41
retropropagación 70, 73
robótica 64, 75
Rosenblatt, Frank 65

S-300, crisis de los misiles 90-91
Searle, John 16
selección de variables 100, 102, 103
Simon, Hebert 26, 28
sistema
  adaptable complejo 114-116, 120-125, 133, 137
  clasificador de Michigan 107
  clasificador de Pittsburg 107
  de mantenimiento de la verdad *véase* TMS
  experto 18, 23, 145
sistemas
  inmunes artificiales 128-131
  simbólicos 26-28
sobreentrenamiento 70, 73, 74, 145
swarm intelligence véase inteligencia de enjambre

teoría de juegos 133-134
Threshold Logic Unit 65
TMS 35
trasplantes de órganos 81-84
Turing, Alan 13, 15, 30
  máquina de 15, 126-127, 144-145
  test de 13-19, 111

Ulam, Stanislaw 125

Van Langren, Michael 104
Vapnik, Vladimir 74, 144
variable latente 103-104, 106, 146
Vaucason, Jacques 112
visualización de datos 104-106
Von Neumann, John 125, 133

Wang, Jing 131
Weizenbaum, Joseph 32, 141

Mentes, máquinas y matemáticas

La inteligencia artificial y sus retos

A menudo vemos en la televisión películas futuristas donde las máquinas son autónomas y capaces de tomar decisiones por ellas mismas. ¿Qué hay de real y de ficción en todo ello? ¿Hasta qué punto está avanzada hoy en día la inteligencia artificial? Este libro propone un viaje fascinante por el futuro de la inteligencia y el rol que las matemáticas juegan en esta apasionante aventura.